EGZAMIN MATURALNY

INFORMATYKA

Poziom rozszerzony

ZBIÓR ZADAŃ

Materiały pomocnicze dla uczniów i nauczycieli

Centralna Komisja Egzaminacyjna
2015

Publikacja opracowana przez zespół koordynowany przez **Renatę Świrko** działający w ramach projektu *Budowa banków zadań* realizowanego przez Centralną Komisję Egzaminacyjną pod kierunkiem Janiny Grzegorek.

Autorzy
dr Lech Duraj
dr Ewa Kołczyk
Agata Kordas-Łata
dr Beata Laszkiewicz
Michał Malarski
dr Rafał Nowak
Rita Pluta
Dorota Roman-Jurdzińska

Komentatorzy
prof. dr hab. Krzysztof Diks
prof. dr hab. Krzysztof Loryś
Romualda Laskowska
Joanna Śmigielska

Opracowanie redakcyjne
Jakub Pochrybniak

Redaktor naczelny
Julia Konkołowicz-Pniewska

Zbiory zadań opracowano w ramach projektu *Budowa banków zadań*,
Działanie 3.2 Rozwój systemu egzaminów zewnętrznych,
Priorytet III Wysoka jakość systemu oświaty,
Program Operacyjny Kapitał Ludzki

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

CENTRALNA
KOMISJA
EGZAMINACYJNA

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Spis treści

Wprowadzenie ... 4
1. Zadania rozwiązywane bez użycia komputera ... 5
   1.1. Analiza algorytmów ... 5
   1.2. Tworzenie algorytmów ... 42
   1.3. Praktyka w teorii ... 72
   1.4. Test z ogólnej wiedzy informatycznej ... 82
2. Zadania praktyczne rozwiązywane z użyciem komputera ... 91
   2.1. Algorytmy w praktyce ... 106
   2.2. Symulacja ... 147
   2.3. Przetwarzanie i tworzenie informacji ... 164
   2.4. Bazy danych ... 185
3. Komentarze do zadań ... 219
4. Odpowiedzi ... 420
5. Wykaz umiejętności ogólnych i szczegółowych sprawdzanych zadaniami ... 504

Wprowadzenie

Prezentowany zbiór zadań z informatyki z rozwiązaniami adresowany jest przede wszystkim do uczniów szkół ponadgimnazjalnych przygotowujących się do egzaminu maturalnego z informatyki w formule obowiązującej od 2015 r.. Zbiór został przygotowany tak, aby można było z niego korzystać zarówno podczas samodzielnej pracy w domu, jak również na lekcjach informatyki pod kierunkiem nauczyciela. W zbiorze znajduje się 110 wiązek zadań, ilustrujących wszystkie rodzaje zagadnień stosowanych na egzaminie maturalnym (w tym zadania pojedyncze oraz zadania składające się z kilku poleceń).

Zbiór składa się z kilku rozdziałów. Rozdział pierwszy i drugi to treści zadań. Zostały one przygotowane w taki sposób, jak są prezentowane zadania na egzaminie maturalnym: rozdział pierwszy to zadania rozwiązywane bez użycia komputera (arkusz I), rozdział drugi to zadania praktyczne — takie, których rozwiązanie wymaga zastosowania znanych narzędzi informatycznych (arkusz II). Rozdział drugi został poprzedzony obszernym wstępem teoretycznym, opisującym sposoby importowania danych do bazy danych, arkusza kalkulacyjnego oraz wczytywania i wypisywania danych w zadaniach programistycznych.

W rozdziale trzecim umieszczono omówienie sposobów rozwiązania zadań, przydatne szczególnie tym uczniom, którzy potrzebują podpowiedzi. Czwarty rozdział zawiera odpowiedzi do wszystkich zadań, a piąty — najistotniejsze sprawdzane zadaniami wymagania ogólne i szczegółowe z *Podstawy programowej*.

Zakres treści i umiejętności sprawdzanych zadaniami jest zgodny z zapisem w podstawie programowej informatyki dla III etapu edukacyjnego (gimnazjum) i IV (szkoła ponadgimnazjalna). Zadania zgrupowano w ośmiu działach tematycznych będących ilustracją wymagań szczegółowych podstawy programowej przedmiotu *Informatyka*. Są to: analiza algorytmów, tworzenie algorytmów, praktyka w teorii, test z ogólnej wiedzy informatycznej, algorytmy w praktyce, symulacja, przetwarzanie i tworzenie informacji, bazy danych.

W zbiorze znajdują się zadania wymagające udzielenia krótkiej odpowiedzi (słowo, kilka słów, zdanie, kilka zdań), zadania reprezentujące różne typy zadań zamkniętych, np.: zadania wyboru wielokrotnego, z luką, zadania typu: prawda-fałsz, a także wiele zadań praktycznych — do rozwiązania z użyciem poznanych narzędzi informatycznych.

Aby ułatwić uczniom przygotowującym się samodzielnie do egzaminu maturalnego korzystanie ze zbioru zadań, w rozdziale trzecim umieszczono komentarze do zadań i dokładne opisy rozwiązań. Komentarze zawierają szczegółowe wskazówki przydatne do rozwiązania zadania, opisy zastosowanych algorytmów, fragmenty kodów programów oraz uwagi, które mogą stanowić podstawę do przemyśleń i samodzielnej pracy nad zadaniami. Z uwagi na rosnącą liczbę zdających, którzy wybierają na egzaminie maturalnym język programowania C/C++, rozwiązania zadań programistycznych przedstawiono w tym właśnie języku.

Mamy nadzieję, że proponowany zbiór zadań będzie pomocny uczniom w przygotowaniu się do egzaminu maturalnego z informatyki, a nauczycielom w monitorowaniu zgodności przebiegu procesu nauczania z obowiązującą *Podstawą programową* przedmiotu informatyka.

Autorzy

1. Zadania rozwiązywane bez użycia komputera

1.1. Analiza algorytmów

Zadanie 1.

Wiązka zadań *Ciągi rekurencyjne*

Dana jest następująca funkcja rekurencyjna:

```
funkcja wynik( i )
    jeżeli i < 3
        zwróć 1 i zakończ;
    w przeciwnym razie
        jeżeli i mod 2 = 0
            zwróć wynik(i – 3) + wynik(i – 1) + 1
        w przeciwnym razie
            zwróć wynik(i – 1) mod 7
```

Uwaga: Operator mod oznacza resztę z dzielenia.

1.1.

Uzupełnij poniższą tabelę:

i	**wynik(*i*)**
2	1
3	
4	
5	
6	
7	
8	

1.2.

Wykonaniem elementarnym nazywać będziemy wykonanie *wynik*(0), *wynik*(1) lub *wynik*(2). Natomiast *złożonością elementarną wynik*(*i*) nazywamy liczbę *wykonań elementarnych* będących efektem uruchomienia *wynik*(*i*). Złożoność elementarną *wynik*(*i*) oznaczamy przez *E*(*i*).

Na przykład złożoność elementarna *wynik*(4) wynosi *E*(4) = 2, ponieważ wykonując *wynik*(4), wywołamy *wynik*(3) i *wynik*(1) (wykonanie elementarne), a z kolei przy wykonaniu *wynik*(3) wywołamy *wynik*(2) (drugie wykonanie elementarne).

Uzupełnij poniższą tabelę:

i	***E*(*i*)**
0	1
3	1
5	
7	
9	
10	

Okazuje się, że $E(i)$ można opisać rekurencyjnym wyrażeniem, którego niekompletną postać podajemy poniżej. Uzupełnij brakujące miejsca tak, aby $E(i)$ dawało poprawną złożoność elementarną *wynik*(*i*) dla każdego całkowitego nieujemnego i.

E(0) = E(1) = E(2) = 1
E(*i*) = E(.....) + E(.....) dla parzystego $i > 2$
E(*i*) = E(.....) dla nieparzystego $i > 2$

1.3.

Naszym celem jest wyznaczenie największej liczby spośród wartości funkcji *wynik*(0), *wynik*(1),…,*wynik*(1000) bez konieczności rekurencyjnego wyznaczania kolejnych wartości. Poniżej prezentujemy niekompletny algorytm realizujący to zadanie.

```
W[0] ← 1
W[1] ← 1
W[2] ← 1
max_wart ← 1
 dla i = 3, 4, …, 1 000 wykonuj
        jeżeli i mod 2 = 0
                W[i] ← .....
        w przeciwnym razie
                W[i] ← .....
        jeżeli W[i] > max_wart
                        .....
 zwróć max_wart
```

Uzupełnij brakujące miejsca w algorytmie tak, aby zwracał on największą liczbę spośród *wynik*(0), *wynik*(1),…,*wynik*(1000).

Zadanie 2.

Wiązka zadań *Ułamki dwójkowe*

W systemach pozycyjnych o podstawie innej niż 10 można zapisywać nie tylko liczby całkowite, ale również rzeczywiste z pewną dokładnością. Na przykład w systemie dwójkowym cyfry po przecinku odpowiadają kolejnym potęgom 1/2 (jednej drugiej). Cyfra 1 na pierwszym miejscu po przecinku odpowiada 1/2, na drugim miejscu — 1/4, na trzecim — 1/8 i tak dalej.

Na przykład $(0{,}101)_2 = 1/2 + 1/8 = 5/8 = 0{,}625_{10}$. Podobnie jak w systemie dziesiętnym nie każda liczba daje się zapisać w ten sposób dokładnie — na przykład liczba 1/3 nie ma skończonego rozwinięcia w systemie dwójkowym (ani też w dziesiętnym). Można jednak stosunkowo łatwo wyznaczyć zadaną liczbę początkowych cyfr po przecinku dla każdej liczby rzeczywistej.

Następujący algorytm przyjmuje na wejściu liczbę rzeczywistą x należącą do przedziału [0, 1) oraz dodatnią liczbę całkowitą k i wypisuje k pierwszych cyfr liczby x w zapisie dwójkowym. Przeanalizuj algorytm i odpowiedz na podane pytania.

Dane:
x — liczba rzeczywista, $0 \le x < 1$,
k — liczba całkowita dodatnia.
Wynik:
zapis dwójkowy liczby x do k-tego miejsca po przecinku.

```
funkcja binarny(x, k)
    wypisz „0,”
    y ← x
    dla i=1, 2, ..., k wykonuj
(*)     jeżeli y ≥ 1/2
            wypisz „1”
        w przeciwnym razie
            wypisz „0”
        y ← y * 2
        jeżeli y ≥ 1
            y ← y – 1
```

2.1.

Podaj liczbę wypisaną przez algorytm dla $x = 0.6$, $k = 5$ oraz wartości zmiennej y przy każdorazowym wykonaniu wiersza oznaczonego (*).

Kolejne wykonanie (*)	**Wartość zmiennej** *y*
1	
2	
3	
4	
5	

Liczba wypisana przez algorytm: .....

2.2.

Podaj przykład liczby *x,* dla której po wykonaniu funkcji *binarny(x,4)* zmienna *y* ma wartość 0, a po wykonaniu funkcji *binarny(x, 3)* zmienna *y* nie jest równa 0.

2.3.

W systemie trójkowym używa się cyfr 0, 1 i 2. Cyfra 1 na pierwszym miejscu po kropce oznacza 1/3, zaś 2 oznacza 2/3. Na drugim miejscu są to odpowiednio 1/9 i 2/9, na trzecim — 1/27 i 2/27 i tak dalej, z kolejnymi potęgami trójki w mianownikach.

Poniżej podany jest algorytm wypisujący dla zadanej liczby rzeczywistej x z przedziału [0,1) oraz liczby całkowitej dodatniej k pierwsze k cyfry zapisu x w systemie trójkowym. Uzupełnij luki tak, aby algorytm działał prawidłowo.

```
funkcja trójkowy(x, k)
        wypisz „0,”
        y ← x
        dla i = 1, 2, ..., k wykonuj
                jeżeli y ≥ 2/3
                        wypisz „2”
                jeżeli .....
                        wypisz „1”
                jeżeli .....
                        wypisz „0”
                y = y * 3
                jeżeli y≥2
                        .....
                jeżeli y≥1
                        .....
```

Zadanie 3.

Wiązka zadań *Ciekawe mnożenia*

Dana jest następująca funkcja rekurencyjna:

Dane:

x — liczba całkowita,

n — dodatnia liczba całkowita.

```
funkcja F(x, n)
     jeżeli n = 1
          podaj wynik x i zakończ
     w przeciwnym razie
          jeżeli n mod 3 = 0
               k ← F(x, n div 3)
(*)            podaj wynik k*k*k i zakończ
          w przeciwnym razie
(**)           podaj wynik x*F(x, n-1) i zakończ
```

Uwaga: „div” jest operatorem dzielenia całkowitego.

3.1.

Podaj wszystkie wywołania rekurencyjne funkcji F oraz obliczany po każdym wywołaniu wynik, jeśli na początku wywołamy F(2, 10).

wywołanie	wynik
F(2, 10)	
F(;)	

3.2.

Uzupełnij tabelę o brakujące elementy:

x	*n*	wynik **F(*x*, *n*)**
2	2	4
2	3	
3		81
	5	32
2		256
	10	1024

3.3.

Uzupełnij tabelę, podając łączną liczbę mnożeń wykonanych w wierszach oznaczonych (*) i (**) po wywołaniu F dla podanych argumentów *x* i *n*:

x	*n*	Liczba operacji mnożenia
2	2	1
2	3	
3	4	
4	7	
4	8	
4	9	

3.4.

Podaj, która z poniższych funkcji określa liczbę wszystkich operacji mnożenia wykonywanych przez powyższy algorytm dla argumentu *n* będącego potęgą trójki ($n = 3^m$ dla pewnego nieujemnego *m*):

- $lmnozen(n) = n \operatorname{div} 2$
- $lmnozen(n) = \log_2 n$
- $lmnozen(n) = 2 \cdot \log_3 n$
- $lmnozen(n) = 1 + \sqrt{n}$

Zadanie 4.

Wiązka zadań *Silniowy system pozycyjny*

Pojęcie *silni* dla liczb naturalnych większych od zera definiuje się następująco:

$$n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (n-1) \cdot n$$

Silniowy system pozycyjny to pozycyjny sposób zapisu liczb naturalnych, w którym mnożniki dla kolejnych pozycji są definiowane przez silnie kolejnych liczb naturalnych, tzn.

$$(x)_! = (x_n x_{n-1}\, x_{n-2} \ldots x_2 x_1)_! = x_n \cdot n! + x_{n-1} \cdot (n-1)! + \cdots + x_2 \cdot 2! + x_1 \cdot 1!$$

W systemie silniowym współczynnik x_i, który odpowiada mnożnikowi $i!$, spełnia zależność $0 \leq x_i \leq i$.

Zapis każdej liczby w silniowym systemie pozycyjnym jest jednoznaczny, tzn. każdą liczbę naturalną można zapisać tylko w jeden sposób i każdą liczbę naturalną można zapisać dokładnie w jeden sposób.

Uwaga: W poniższych zadaniach będziemy mieć do czynienia tylko z takimi liczbami, dla których współczynniki x_i spełniają zależność $0 \leq x_i \leq 9$.

Przykład

$$(1220)_! = 1 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 0 \cdot 1! = 24 + 12 + 4 + 0 = 40.$$

4.1.

Uzupełnij tabelę. Zamień zapis liczby w systemie silniowym na jej zapis w systemie dziesiętnym.

liczba w systemie silniowym	**liczba w systemie dziesiętnym**
$(310)_!$	
$(2011)_!$	
$(54211)_!$	

4.2.

Podaj zapis w systemie silniowym największej liczby, jaką można w tym systemie zapisać na pięciu pozycjach.

4.3.

Zamiana zapisu liczby w systemie dziesiętnym na zapis w systemie silniowym może przebiegać według następującego schematu: Szukamy największej liczby *k*, której silnia nie przekracza liczby *x*. Pierwsza jej cyfra to wynik dzielenia całkowitego *x* przez *k*!. Kolejne cyfry zapisu silniowego (zaczynając od cyfr najbardziej znaczących) otrzymujemy przez wyznaczanie wyników dzielenia liczby *x* przez (*k* – 1)!, (*k* –2)!, ..., 2!, 1!. Po wyznaczeniu cyfry x_i, odpowiadającej współczynnikowi *i*!, zmniejszamy wartość *x* o liczbę odpowiadającą cyfrze x_i, czyli $x_i \cdot i!$. Oznacza to, że *x* przyjmuje wartość *x* mod *k*!.

Przykład

x	***k***	***x* div *k*!**	***x* mod *k*!**
1548	6	2	**108**
108	5	0	108
108	4	4	12
12	3	2	0
0	2	0	0
0	1	0	0

Liczba dziesiętna 1548 w zapisie silniowym: $(204200)_!$

Wykonaj zamianę liczby 5489 z systemu dziesiętnego na silniowy zgodnie z opisanym powyżej algorytmem. Uzupełnij poniższą tabelkę oraz podaj zapis silniowy liczby 5489.

x	***k***	***x* div *k*!**	***x* mod *k*!**
5489			

Liczba dziesiętna 5489 w zapisie silniowym: **.....**

4.4.

Poniżej przedstawiono algorytm z lukami, który zamienia zapis liczb z systemu dziesiętnego na system silniowy. Uzupełnij luki w tym algorytmie.

Specyfikacja

Dane:

x — liczba całkowita dodatnia zapisana w systemie dziesiętnym,

Wynik:

s — napis reprezentujący liczbę *x* zapisaną w systemie silniowym.

silnia ← 1
k ←1
dopóki (*silnia* < *x*) **wykonuj**
    k ← *k* + 1
    silnia← *silnia* k*
jeżeli **.....**
    silnia ← *silnia* div *k*
    k ← *k* – 1
s ←" "
dopóki (*k*>0) **wykonuj**
    cyfra ← **.....**
    s ← *s* ◦ tekst (*cyfra*)
    x ← **.....**
    silnia ← **.....**
    k ←*k* – 1

Uwaga
tekst (*x*) oznacza funkcję zamieniającą liczbę *x* na jej zapis tekstowy
" " oznacza napis pusty
u ◦ *v* oznacza sklejenie dwóch napisów: *u* oraz *v*

Zadanie 5.

Wiązka zadań *Sortowanie przez wstawianie na dwa sposoby*

Sortowanie przez wstawianie polega na powtarzaniu operacji wstawiania elementu do już uporządkowanego ciągu. Aby znaleźć w uporządkowanym ciągu miejsce, w które należy wstawić nowy element, można stosować różne strategie. Poniższy algorytm znajduje to miejsce metodą wyszukiwania binarnego.

Specyfikacja

Dane:

n — liczba naturalna oznaczająca długość ciągu,

A[1..n] — ciąg liczb całkowitych zapisanych w tablicy

Wynik:

A[1..n] — tablica liczb całkowitych, w której liczby zostały ustawione w porządku niemalejącym

Algorytm

```
dla j = n – 1, n – 2, … , 1 wykonuj
    x ← A[j]
    p ← j
    k ← n + 1
    dopóki k – p > 1 wykonuj
        i ← (p + k) div 2
        jeżeli x ≤ A[i]
            k ← i
        w przeciwnym razie
            p ← i
    dla i = j, j + 1, … , p – 1 wykonuj
        A[i] ← A[i + 1]
    A[p] ← x
```

5.1.

Przeanalizuj działanie powyższego algorytmu dla ciągu 12, 4, 3, 10, 5, 11 o długości $n = 6$ i podaj, ile razy zostaną wykonane instrukcje $k \leftarrow i$ i $p \leftarrow i$ dla kolejnych wartości j zamieszczonych w tabeli.

Wartość *j*	**Liczba wykonań**	
	k* ← *i	***p* ← *i***
5		
4		
3		
2		
1		

5.2.

Uzupełnij luki w poniższym algorytmie sortowania przez wstawianie tak, aby znajdowanie miejsca na kolejny wstawiany element było realizowane metodą wyszukiwania liniowego.

Specyfikacja

Dane:

n — liczba naturalna oznaczająca długość ciągu, A[1..n] — ciąg liczb całkowitych zapisanych w tablicy.

Wynik:

A[1..n] — tablica liczb całkowitych, w której liczby zostały ustawione w porządku niemalejącym.

Algorytm:

dla $j = n - 1, n - 2, \ldots, 1$ **wykonuj**
  $x \leftarrow \ldots\ldots\ldots$
  $i \leftarrow \ldots\ldots\ldots$
  dopóki $(i \leq n)$ **i** $(x >$ A[i]) **wykonuj**
    $A[i - 1] \leftarrow A[i]$
    $i \leftarrow i + 1$
  $\ldots\ldots \leftarrow x$

5.3.

Porównaj dwa algorytmy sortowania przez wstawianie: taki, w którym miejsce dla wstawianego elementu jest znajdowane metodą wyszukiwania binarnego, i taki, w którym jest ono znajdowane metodą wyszukiwania liniowego. Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe (P), a które jest fałszywe (F).

Oba algorytmy dla ciągu 12, 4, 3, 10, 5, 11 wykonują

	P	F
jednakową liczbę porównań między elementami ciągu liczb.		
jednakową liczbę przesunięć elementów w tablicy.		
tyle samo powtórzeń pętli zewnętrznej w algorytmie.		
jednakową liczbę instrukcji podstawienia wartości do zmiennej *x*.		

Zadanie 6.

Wiązka zadań *Od szczegółu do ogółu*

Rozważmy następujący algorytm:

Dane:

k — liczba naturalna,

A[1...2^k] — tablica liczb całkowitych.

Algorytm 1:

```
    n ← 1
    dla i=1,2,…,k wykonuj
        n ← 2·n
    s ← 1
    dopóki s<n wykonuj
        j ← 1
        dopóki j<n wykonuj
(*)         jeżeli A[j]>A[j+s]
(**)            zamień(A[j],A[j+s])
            j ← j+2·s
        s ← 2·s
    zwróć A[1]
```

Uwaga: Funkcja *zamień*(*A*[*j*],*A*[*j*+*s*]) zamienia miejscami wartości *A*[*j*] oraz *A*[*j*+*s*].

6.1.

Prześledź działanie algorytmu 1 dla podanych poniżej wartości *k* i początkowych zawartości tablicy *A*. W każdym wierszu poniższej tabeli wpisz końcową zawartość tablicy *A*.

k	Początkowa zawartość tablicy A[1...2^k]	Końcowa zawartość tablicy A[1...2^k]
2	[4, 3, 1, 2]	[1, 4, 3, 2]
2	[2, 3, 4, 1]	
3	[1, 2, 3, 4, 5, 6, 7, 8]	
3	[8, 7, 6, 5, 4, 3, 2, 1]	
3	[4, 5, 6, 1, 8, 3, 2, 4]	

6.2.

Wskaż, które z poniższych zdań są prawdziwe (P), a które fałszywe (F), wstawiając znak X w odpowiedniej kolumnie:

	P	F
Po zakończeniu działania algorytmu 1 komórka $A[2^k]$ zawiera największą z liczb $A[1],\ldots,A[2^k]$.		
Po zakończeniu działania algorytmu 1 spełniona jest nierówność $A[i] \leq A[i+1]$ dla każdego i, takiego że $1 \leq i \leq 2^k$.		
Po zakończeniu działania algorytmu 1 komórka $A[1]$ zawiera najmniejszą z liczb $A[1],\ldots,A[2^k]$.		

6.3.

Wskaż, które z poniższych zdań są prawdziwe (P), a które fałszywe (F), wstawiając znak X w odpowiedniej kolumnie. Przyjmij, że $n=2^k$ oraz $k>1$:

	P	F
Instrukcja jeżeli w wierszu (*) jest wykonywana mniej niż $2n$ razy.		
Instrukcja jeżeli w wierszu (*) jest wykonywana mniej niż $n/2$ razy.		
Możliwe jest dobranie takiej początkowej zawartości $A[1..2^k]$, że instrukcja zamiany z wiersza (**) nie zostanie wykonana ani razu.		
Możliwe jest dobranie takiej początkowej zawartości $A[1..2^k]$, że instrukcja zamiany z wiersza (**) zostanie wykonana co najmniej $2n^2$ razy.		

6.4.

Rozważmy poniższy algorytm podobny do **algorytmu 1**.

Wejście: k — liczba naturalna,
$A[1...2^k]$ — tablica liczb całkowitych.

Algorytm 2:

```
    n ← 1
    dla i=1,2,…,k wykonuj
        n ← 2·n
    s ← 1
    dopóki s<n wykonuj
        j ← 1
        dopóki j<n wykonuj
(*)         jeżeli A[j]>A[j+1]
                zamień(A[j],A[j+1])
            j ← j+1
        s ← s+1
    zwróć A[1], A[2],…,A[n]
```

Uwaga: Funkcja *zamień*($A[j]$,$A[j+1]$) zamienia miejscami wartości $A[j]$ oraz $A[j+1]$.

Uzupełnij luki w poniższych zdaniach. Przyjmij $n=2^k$ oraz $k>1$.

Po zakończeniu działania algorytmu 2 element $A[i]$ jest .....
niż element $A[i+1]$ dla każdego i większego od .....
oraz mniejszego od.....

Wiersz (*) **algorytmu 2** wykonywany będzie w przebiegu algorytmu

- ..... niż n razy,
- ..... niż n^2 razy.

Zadanie 7.

Wiązka zadań *Scalanie*

Rozważmy następujący algorytm, który jako dane przyjmuje tablicę *n*-elementową, gdzie *n* jest potęgą dwójki:

Dane: tablica liczb rzeczywistych T[1..*n*], gdzie $n = 2^m$, a *m* jest liczbą całkowitą nieujemną

```
funkcja uporządkuj(T[1..n]):
    jeżeli n=1
          zwróć T[1..n] i zakończ
    k ← n/2
    A[1..k] ← uporządkuj(T[1 .. k])
    B[1..k] ← uporządkuj(T[k+1 .. n])
    zwróć scal(A, B) i zakończ
```

Funkcja *scal*(*A*,*B*) dla danych dwóch tablic o rozmiarze *k* zwraca tablicę o rozmiarze *2k*, powstałą przez połączenie tablic *A* i *B* w sposób uporządkowany, tj. od elementu najmniejszego do największego. Na przykład dla tablic *A* = [4,6,18,22] i *B* = [1,3,10,15] wywołanie *scal*(*A*,*B*) zwróci tablicę [1,3,4,6,10,15,18,22].

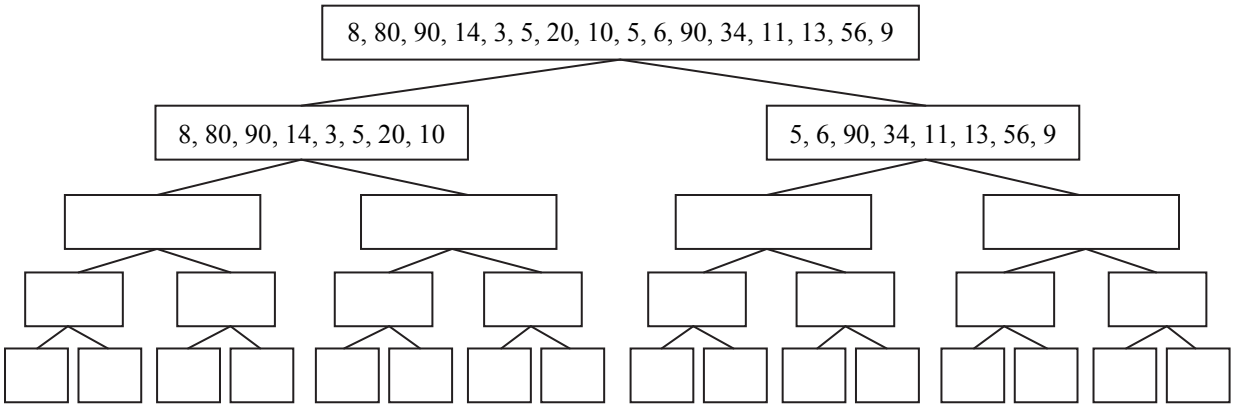
7.1.

Spośród danych tablic wybierz te, które są zgodne ze specyfikacją algorytmu, a następnie podaj dla nich wynik jego działania.

- *T* = [15, 11],
- *T* = [1, 3, 8],
- *T* = [8, 4, 2, 1],
- *T* = [10, 15, 1 , 6, 9, 2, 5, 90].

7.2.

Funkcja *uporządkuj* jest funkcją rekurencyjną. Uzupełnij poniższe drzewo wywołań rekurencyjnych dla danej tablicy *T* = [8, 80, 90, 14, 3, 5, 20, 10, 5, 6, 90, 34, 11, 13, 56, 9].



7.3.

Załóżmy, że wywołanie procedury *scal*(*A*,*B*) dla dwóch tablic o długości *k* wykonuje *2k*–1 kosztownych operacji (porównywania liczb). Podaj liczbę kosztownych operacji, jaka zostanie wykonana przez funkcję *uporządkuj* dla tablicy

$$T[1..16] = [\ 8, 80, 90, 14, 3, 5, 20, 10, 5, 6, 90, 34, 11, 13, 56, 9\ ].$$

Zadanie 8.

Wiązka zadań *Dwa ciągi*

Niech $A[1..n]$ i $B[1..n]$ będą uporządkowanymi rosnąco tablicami liczb całkowitych i niech x będzie liczbą całkowitą. Rozważmy następujący algorytm:

```
    i←1
    j←n
    dopóki  (i ≤ n oraz j > 0) wykonuj

    dopóki (i ≤ n  oraz  j > 0)wykonuj
(*)     jeżeli A[i] + B[j] = x
            podaj wynik PRAWDA i zakończ algorytm
        w przeciwnym razie
                jeżeli A[i] + B[j] < x
                        i← i + 1
                w przeciwnym razie
                        j← j – 1
    podaj wynik FAŁSZ
```

8.1.

Uzupełnij poniższą tabelę. Podaj wynik działania algorytmu oraz liczbę porównań wykonanych w wierszu oznaczonym (*).

Tablica *A*	**Tablica *B***	***x***	**Wynik działania algorytmu**	**Liczba porównań w kroku (*)**
3, 5, 12, 17	8, 10, 13, 14	21		
4, 6, 8, 10	5, 7, 9, 11	13		

8.2.

Przeanalizuj działanie zaprezentowanego algorytmu i uzupełnij poniższą specyfikację.

Dane:

n — dodatnia liczba całkowita
A[1..*n*], *B*[1..*n*] — *n*-elementowe tablice liczb całkowitych, posortowane rosnąco
x — liczba całkowita

Wynik:

PRAWDA, gdy .....

FAŁSZ, gdy .....

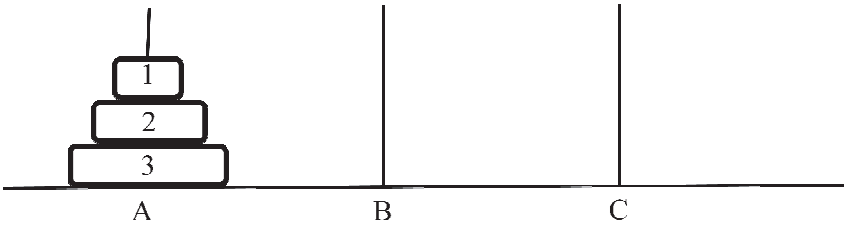
8.3.

Podaj przykład pięcioelementowych tablic *A* i *B*, dla których przy $x = 20$ algorytm wykona sześć porównań w wierszu (*), a wynikiem będzie *PRAWDA*.

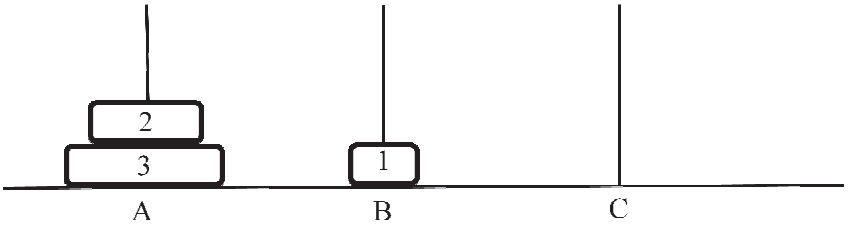
Zadanie 9.

Wiązka zadań *Wieże z Hanoi*

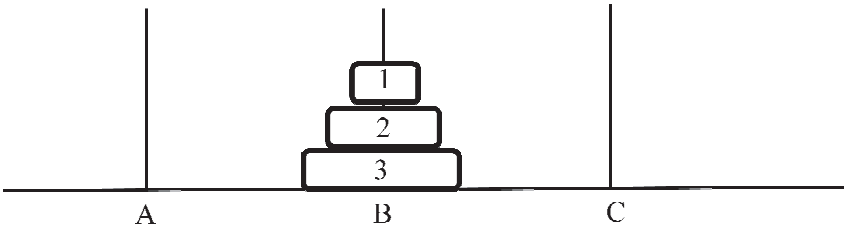
W problemie wież z Hanoi mamy trzy pręty oznaczone A, B i C oraz n okrągłych krążków o średnicach odpowiednio 1, 2, …, n. Na początku wszystkie krążki nałożone są na pręt A, w kolejności od największego do najmniejszego (największy na dole, najmniejszy na górze). Układ ten (dla n=3) został przedstawiony na poniższym rysunku.



Zgodnie z regułami problemu krążki można przekładać między prętami. W jednym ruchu możliwe jest przełożenie krążka znajdującego się na szczycie jednego z prętów na szczyt innego pręta, pod warunkiem że nie kładziemy przekładanego krążka na krążek mniejszy od niego. Na przykład na poniższym rysunku krążek 2 możemy przełożyć z pręta A na pręt C, natomiast niemożliwe jest przełożenie go na pręt B.



Zadanie polega na przełożeniu wszystkich krążków z pręta A na pręt B, przy czym można korzystać z pomocniczego pręta C. Na poniższym rysunku przedstawiono efekt końcowy.



Problem wież z Hanoi można rozwiązać za pomocą algorytmu rekurencyjnego. W algorytmie pręty: startowy, docelowy i pomocniczy, podane są jako parametry wejściowe, odpowiednio x, y i z. Algorytm polega na tym, że najpierw przenosimy $n - 1$ krążków na pręt pomocniczy z, potem największy krążek zostaje przeniesiony na pręt docelowy y, a na koniec $n - 1$ krążków zostaje przeniesionych z pręta pomocniczego z na pręt docelowy y, przy czym pręt startowy x traktowany jest jako pomocniczy.

Algorytm

Specyfikacja

Dane:

n — liczba całkowita dodatnia,

x — nazwa pręta startowego,

y — nazwa pręta docelowego,

z — nazwa pręta pomocniczego.

Wynik:

ciąg ruchów opisujący rozwiązanie problemu wież z Hanoi z n krążkami, w którym na początku wszystkie krążki znajdują się na pręcie x, a na końcu mają znaleźć się na pręcie y, zaś pomocniczym prętem jest z.

Uwaga: Pojedynczy ruch zapisujemy za pomocą znaku $\Rightarrow$. Na przykład C $\Rightarrow$ B oznacza przeniesienie krążka z pręta C na pręt B.

```
funkcja wieże(n, x, y, z)
    jeżeli n=1
        wypisz x ⇒ y
    w przeciwnym razie
        wieże(n – 1, x, z, y)
        wypisz x ⇒ y
        wieże(n – 1, z, y, x)
```

Przykład

Wywołanie *wieże*(2, A, B, C) spowoduje dwa wywołania rekurencyjne: *wieże*(1, A, C, B) oraz wieże(1, C, B, A). Ciąg ruchów utworzony przez *wieże*(2, A, B, C) ma postać:

$$A \Rightarrow C, A \Rightarrow B, C \Rightarrow B,$$

gdzie podkreślone ruchy są utworzone przez rekurencyjne wywołania *wieże*(1, A, C, B) oraz *wieże*(1, C, B, A).

9.1.

Podaj wszystkie wywołania rekurencyjne funkcji *wieże* (wraz z ich parametrami), do których dojdzie w wyniku wywołania *wieże*(3, A, B, C). Odpowiedź podaj w poniższej tabeli, uzupełniając parametry wszystkich wywołań rekurencyjnych.

n	*x*	*y*	*z*
3	A	B	C
2	A	C	B
1	A	B	C
1			
2			
1			
1			

9.2.

Prześledź działanie *wieże*(3, A, B, C) i uzupełnij poniżej wygenerowany ciąg ruchów:

A⇒B; A⇒C; ……

9.3.

Niech H(*n*) oznacza liczbę ruchów wykonanych przez podany algorytm dla *n* krążków. Zauważ, że rozwiązanie problemu dla *n*>1 krążków wymaga jednego ruchu oraz dwukrotnego rozwiązania problemu dla *n* – 1 krążków. W oparciu o tę obserwację uzupełnij poniższą tabelę.

n	H(*n*)
1	1
2	3
3	
4	
5	
7	
10	

Podaj ogólny wzór określający liczbę ruchów dla *n* krążków:

H(*n*)= .....

9.4.

Poniżej prezentujemy nierekurencyjne rozwiązanie problemu wież z Hanoi.

Specyfikacja

Dane:

n — liczba całkowita dodatnia,

Wynik:

ciąg ruchów opisujący rozwiązanie problemu wież z Hanoi z n krążkami, w którym na początku wszystkie krążki znajdują się na pręcie A, a na końcu powinny się znaleźć na pręcie B.

Algorytm:

(1) **dopóki** (pręt A jest niepusty lub pręt C jest niepusty) **wykonuj**
(2) **jeżeli** n jest parzyste:
(3) **przenieś** krążek nr 1 o jedną pozycję w lewo
(4) **w przeciwnym razie**
(5) **przenieś** krążek nr 1 o jedną pozycję w prawo
(6) **przenieś** krążek między prętami, na których nie ma krążka nr 1

W powyższym algorytmie przeniesienie krążka nr 1 o jedną pozycję w prawo oznacza wykonanie jednego z ruchów A⇒B, B⇒C lub C⇒A, tak aby krążek nr 1 został przeniesiony na inny pręt. Analogicznie przeniesienie krążka w lewo oznacza wybranie jednego z ruchów A⇒C, B⇒A lub C⇒B, tak aby krążek nr 1 został przeniesiony na inny pręt.

Ruch w kroku (6) powyższego algorytmu jest określony jednoznacznie, gdyż dopuszczalne jest tylko położenie mniejszego krążka na większym, a nie odwrotnie.

Przykład

Dla *n*=3 powyższy algorytm wykona następujący ciąg ruchów:

A⇒B; A⇒C; B⇒C; A⇒B; C⇒A; C⇒B; A⇒B,

gdzie ruchy podkreślone przenoszą krążek nr 1 o jedną pozycję w prawo.

Wypisz ciąg ruchów, który poda powyższy algorytm dla *n*=4.

Zadanie 10.

Wiązka zadań *Dzielenie wielomianu*

Rozważamy wielomiany $P(x)$ stopnia $n-1$, gdzie n jest potęgą dwójki:

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}, \qquad n = 2^m.$$

Do obliczania wartości $P(x)$ można zastosować technikę „dziel i zwyciężaj” w następujący sposób: dzielimy postać ogólną wielomianu $P(x)$ na dwie części:

$$P(x) = \underbrace{a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}}_{\text{cz. I}} + \underbrace{a_k x^k + a_{k+1} x^{k+1} + \cdots + a_{n-1} x^{n-1}}_{\text{cz. II}},$$

gdzie $k = n/2$. Jeśli w drugiej części wydzielimy czynnik x^k, to otrzymamy równość

(*) $$P(x) = A(x) + B(x) \cdot x^k,$$

gdzie $A(x)$ i $B(x)$ są wielomianami stopnia $k-1$:

$$A(x) = a_{k-1} x^{k-1} + \cdots + a_2 x^2 + a_1 x + a_0,$$
$$B(x) = a_{n-1} x^{k-1} + \cdots + a_{k+2} x^2 + a_{k+1} x + a_k,$$

W ten sposób problem obliczenia wartości $P(x)$ dzielimy na dwa podproblemy — obliczenia $A(x)$ i obliczenia $B(x)$, przy czym każdy z nich ma rozmiar o połowę mniejszy. Na przykład aby obliczyć wartość wielomianu ($n = 8$)

$$P(x) = -8 + 7x + 6x^2 - x^3 + 4x^4 + 5x^5 - 2x^6 + 3x^7,$$

obliczamy $k = n/2 = 4$, $A(x) = -8 + 7x + 6x^2 - x^3$ oraz $B(x) = 4 + 5x - 2x^2 + 3x^3$. Następnie korzystamy ze wzoru $P(x) = A(x) + B(x) \cdot x^4$.

Ponieważ do pełnej realizacji algorytmu we wzorze (*) potrzebne jest obliczenie wartości x^k, więc najpierw obliczamy wszystkie potęgi $x, x^2, x^4, \ldots, x^{2^{m-1}}$ i zapamiętujemy je w tablicy Z[0 .. m-1], np. za pomocą poniższej procedury.

Obliczanie tablicy $Z[0..m\text{-}1]$:

Dane:

n — liczba całkowita postaci 2^m, gdzie m jest liczbą całkowitą nieujemną,

x — liczba rzeczywista.

Wynik:

tablica $Z[0 .. m\text{-}1]$, dla której $Z[j] = x^{2^j}$

$Z[0] \leftarrow x$
$m \leftarrow 1$
$w \leftarrow 2$
dopóki $w < n$ **wykonuj**
    $Z[m] \leftarrow Z[m-1] \cdot Z[m-1]$
    $m \leftarrow m + 1$
    $w \leftarrow 2 \cdot w$

Mając tablicę Z, obliczamy wartość $P(x)$ za pomocą funkcji rekurencyjnej *F*.

Dane:

tablica liczb rzeczywistych $T[0\,..\,n-1]$ *ze współczynnikami* $a_0, a_1, \ldots, a_{n-1}$, *gdzie* $n = 2^m$, *a m jest liczbą całkowitą nieujemną;*

tablica Z[0 .. m-1], dla której $Z[j] = x^{2^j}$.

Wynik:

wartość $a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0$.

funkcja F($T[0\,..\,n-1]$):
  jeżeli *n*=1
    zwróć T[0] **i zakończ**
  $k \leftarrow n/2$
  $A[0\,..\,k-1] \leftarrow T[0\,..\,k-1]$
  $B[0\,..\,k-1] \leftarrow T[k\,..\,n-1]$
  zwróć $\mathrm{F}(A) + \mathrm{F}(B) \cdot Z[m-1]$ **i zakończ**

Prześledźmy na przykład obliczenie wartości wielomianu

$$P(x) = -8 + 7x + 6x^2 - x^3 + 4x^4 + 5x^5 - 2x^6 + 3x^7 \qquad (n = 2^m, \qquad m = 3)$$

dla *x*=3.

W pierwszym kroku algorytm obliczy tablicę $Z[0\,..\,2]$:

$$Z[0] = 3, \qquad Z[1] = 3^2, \qquad Z[2] = 3^4 = 81.$$

Następnie algorytm obliczy *F*([-8,7,6,-1,4,5,-2,3]), tzn.

F([-8,7,6,-1,4,5,-2,3]) = *F*([-8,7,6,-1]) + *F*([4,5,-2,3])*Z[2]

Obliczanie *F*([4,5,-2,3]) i *F*([-8,7,6,-1]) odbywa się w analogiczny sposób, tzn.

F([-8,7,6,-1]) = *F*([-8,7])+ *F*([6,-1])*Z[1]
= F([-8])+F([7])*Z[0]+(F([6])+F([-1])*Z[0])*Z[1]
= (-8)+7*Z[0] + (6+(-1)*Z[0])*Z[1]
= (-8)+7*3+(6+(-1)*3)*9 = 40,

F([4,5,-2,3]) = *F*([4,5])+*F*([-2,3])*Z[1]
= *F*([4])+*F*([5])*Z[0]+(*F*([-2])+*F*([3])*Z[0])*Z[1]
= 4+5*Z[0]+((-2)+3*Z[0])*Z[1]
= 4+5*3+((-2)+3*3)*9=82.

Zatem

$$F([-8,7,6,-1,4,5,-2,3]) = F([-8,7,6,-1])+F([4,5,-2,3])*Z[2] = 40+82*81 = 6682.$$

Można zauważyć, że podczas obliczania $F([-8,7,6,-1,4,5,-2,3])$ łącznie zostanie wykonanych 7 mnożeń:

- 3 mnożenia podczas obliczania $F([-8,7,6,-1])=40$,
- 3 mnożenia podczas obliczania $F([4,5,-2,3])=82$,
- 1 mnożenie przy obliczaniu $F([-8,7,6,-1])+ F([4,5,-2,3])*Z[2] = 40+82*Z[2]$.

Liczba wszystkich wywołań rekurencyjnych jest równa 15. Oto wszystkie wywołania funkcji F:

- $F([-8,7,6,-1,4,5,-2,3])$
  - $F([-8,7,6,-1])$
    - $F([-8,7])$
      - $F([-8])$
      - $F([7])$
    - $F([6,-1])$
      - $F([6])$
      - $F([-1])$
  - $F([4,5,-2,3])$
    - $F([4,5])$
      - $F([4])$
      - $F([5])$
    - $F([-2,3])$
      - $F([-2])$
      - $F([3])$

10.1.

Podaj, jakie wartości zostaną obliczone w tablicy $Z[0 .. m-1]$ przy obliczaniu $P(2)$, gdzie

$$P(x) = 9 + x - 6x^3 + 2x^7 - 3x^{14} + 5x^{15}.$$

10.2.

Oblicz łączną liczbę operacji mnożenia, jaka zostanie wykonana przez funkcję F(T) dla n-elementowej tablicy T.

Uzupełnij poniższą tabelkę:

n	**Liczba operacji mnożenia**
1	0
2	
4	
8	
16	
1024	

Uzupełnij poniższy wzór rekurencyjny dla $f(n)$ — liczby operacji mnożenia wykonywanych przez funkcję F dla tablicy n-elementowej:

$$f(n) = \ldots\ldots\ldots \cdot f(n/2) + \ldots\ldots\ldots\ldots \quad \text{dla } n > 1$$

10.3.

Wypisz wszystkie wywołania funkcji *F* podczas obliczania

$$P(x) = 9 + x - 6x^3 + 10x^5 + 2x^7 + 4x^9 - x^{10} - 3x^{14} + 5x^{15}.$$

Podaj wzór ogólny na *g*(*n*) — liczbę wszystkich wywołań funkcji *F* dla wielomianu $P(x)$ stopnia $n-1$, gdzie n jest potęgą dwójki. Uzupełnij poniższy wzór:

$$g(n) =.....$$

10.4.

Do obliczania wartości wielomianu $P(x)$

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \qquad (n = 3^m),$$

a więc gdy n jest potęgą trójki, można zastosować podział na trzy części

$$P(x) = \underbrace{a_0 + \cdots + a_{k-1} x^{k-1}}_{\text{cz. I}} + \underbrace{a_k x^k + \cdots + a_{2k-1} x^{2k-1}}_{\text{cz. II}} + \underbrace{a_{2k} x^{2k} + \cdots + a_{n-1} x^{n-1}}_{\text{cz. III}},$$

gdzie $k = n/3$. Wówczas uzyskujemy

$$P(x) = A(x) + B(x) \cdot x^k + C(x) \cdot x^{2k} = A(x) + (B(x) + C(x) \cdot x^k) \cdot x^k,$$

gdzie $A(x), B(x), C(x)$ są wielomianami, w których liczba współczynników jest trzykrotnie mniejsza niż w wyjściowym wielomianie $P(x)$.

Wzorując się na funkcji *F*, możemy skonstruować rekurencyjną funkcję *G*(T[0 .. *n*-1]), obliczającą wartość $P(x)$ przez podział tablicy *T* na trzy równe części, o ile $n > 1$. Po obliczeniu trzech wyników dla każdej z części, powiedzmy $A(x), B(x), C(x)$, funkcja oblicza swój wynik, wykonując dodatkowo dwa mnożenia:

- jedno mnożenie przy obliczaniu $C(x) \cdot x^k$.
- jedno mnożenie przy obliczaniu $(B(x) + C(x) \cdot x^k) \cdot x^k$.

Podobnie jak poprzednio pomijamy problem obliczania potęg x^k dla $k = 3^j$ $(j \geq 0)$, tzn. przyjmujemy, że potęgi te są przechowywane w pewnej tablicy *Z*[0 .. *m*-1], a więc ich obliczanie nie jest wliczane do kosztu obliczeniowego funkcji *G*.

W ten sposób na przykład dla *n*=3 funkcja *G* wykona dokładnie 2 mnożenia, a dla *n*=9 funkcja wykona łącznie 8 mnożeń:

- po 2 mnożenia dla każdej z trzech części $A(x), B(x), C(x)$,
- jedno mnożenie przy obliczaniu $C(x) \cdot x^k$.
- jedno mnożenie przy obliczaniu $(B(x) + C(x) \cdot x^k) \cdot x^k$.

Uzupełnij poniższą tabelkę, podając liczbę mnożeń, jaka zostanie wykonana przez funkcję *G* dla *n*-elementowej tablicy współczynników T.

n	**Liczba operacji mnożenia**
3	2
9	8
27	
81	
243	

Zadanie 11.

Wiązka zadań *Odwrotna notacja polska*

Dla przetwarzania przez komputer wygodnym sposobem zapisu wyrażeń arytmetycznych jest tzw. odwrotna notacja polska (ONP). Zapis w ONP wyrażenia W nazywamy *postacią ONP* i oznaczamy ją ONP(W). W ONP operator (dodawania, odejmowania, mnożenia, dzielenia) umieszczamy za jego argumentami, np. 2+5 zapisujemy jako 2 5 +. Dokładniej, postać ONP dla wyrażenia definiujemy rekurencyjnie w następujący sposób:

1. Jeżeli W jest liczbą, to jego postać ONP jest równa W.
2. Jeżeli W ma postać W_1 *op* W_2, gdzie *op* jest operatorem, a W_1 i W_2 wyrażeniami, to ONP(W) jest równe ONP(W_1) ONP(W_2) *op*.

Przykład

$W = W_1$ *op* W_2	W_1	W_2	*op*	ONP(W)
1 + 2	1	2	+	1 2 +
5 – 7	5	7	–	5 7 –
3 ∗ (5 – 7)	3	5 – 7	∗	3 5 7 – ∗
(1 + 2) + (3 ∗ (5 – 7))	1 + 2	3 ∗ (5 – 7)	+	1 2 + 3 5 7 – ∗ +

Zauważmy, że dla W= (1 + 2) + (3 ∗ (5 – 7)) wartość ONP(W) uzyskujemy z połączenia ONP(1 + 2) = 1 2 +, ONP(3 ∗ (5 – 7)) = 3 5 7 – ∗ oraz znaku dodawania +.

11.1.

Uzupełnij poniższą tabelę, podając dla każdego wyrażenia z pierwszej kolumny jego podwyrażenia, łączący je operator oraz postać ONP tego wyrażenia.

$W = W_1$ *x* W_2	W_1	W_2	*op*	ONP(W)
4 + 3	4	3	+	4 3 +
(4 + 3) ∗ 2				
5 ∗ (7 – 6)				
((4 + 3) ∗ 2) – (5 ∗ (7 – 6))				

11.2.

Postać ONP wyrażeń, choć dla ludzi mało czytelna, ma własności bardzo przydatne dla analizy komputerowej. W ONP nie są potrzebne nawiasy, a do wyznaczania wartości wyrażenia można zastosować prosty algorytm podany poniżej.

Specyfikacja

Dane:

n — liczba całkowita dodatnia,

$X = X[1 \ldots n]$ — wyrażenie w ONP, gdzie $X[i]$ dla $1 \leq i \leq n$ jest liczbą lub znakiem ze zbioru $\{+, -, *\}$.

Wynik:

wartość wyrażenia X.

Algorytm:

```
k ← 1
dla i=1,2,…,n wykonuj
    jeżeli X[i] jest liczbą
        T[k] ← X[i]
    jeżeli X[i]∈ {+, –, *}
        b ← T[k – 1]
        a ← T[k – 2]
        k ← k – 2
        jeżeli X[i] = `+`
            T[k] ← a + b
        jeżeli X[i] = `–`
            T[k] ← a – b
        jeżeli X[i] = `*`
            T[k] ← a * b
    k ← k + 1
zwróć T[1]
```

Prześledź działanie podanego algorytmu dla wyrażenia X = 9 7 + 3 * 5 4 – 2 * –, czyli dla n=11 oraz następujących wartości $X[1],.., X[11]$:

i	1	2	3	4	5	6	7	8	9	10	11
$X[i]$	9	7	+	3	*	5	4	–	2	*	–

Uzupełnij poniższą tabelę:

i	Wartość zmiennej k po i-tym przebiegu pętli	Zawartość tablicy $T[1..k-1]$ po i-tym przebiegu pętli
1	2	9
2	3	9, 7
4	3	16, 3
5		
6		
10		
11		

11.3.

Poniższy algorytm sprawdza, czy podany na wejściu ciąg liczb i operatorów jest poprawnym wyrażeniem w ONP.

Specyfikacja

Dane:

n — liczba całkowita dodatnia

$X = X[1..n]$ — ciąg elementów, z których każdy jest liczbą lub znakiem ze zbioru $\{+, -, *\}$.

Wynik:

Tak — jeśli X jest poprawnym wyrażeniem w ONP,

Nie — w przeciwnym przypadku.

Algorytm:

```
licznik ← 0
dla i=1,2,…,n wykonuj
    jeżeli X[i] jest liczbą
        licznik ← licznik + 1
    jeżeli X[i]∈ {+, –, *}
        jeżeli licznik < 2
                zwróć „Nie” i zakończ działanie
        w przeciwnym razie
                licznik ← licznik – 1
jeżeli licznik ≠ 1
    zwróć „Nie”
w przeciwnym razie
    zwróć „Tak”
```

Oceń, które z podanych poniżej napisów są wyrażeniami zapisanymi poprawnie w ONP, wpisując słowa TAK lub NIE w trzeciej kolumnie poniższej tabeli. W drugiej kolumnie podaj wartości zmiennej *licznik* po zakończeniu działaniu algorytmu dla poszczególnych napisów.

Napis	Wartość zmiennej *licznik* po zakończeniu algorytmu	Czy poprawne wyrażenie w ONP?
1 2 + ∗	1	NIE
1 2 + 3 4 – 5 ∗ 7 8 + 9		
1 2 3 4 5 + + + +		
1 2 3 4 5 + + + + + +		
1 2 3 4 5 + + + + +		
1 2 + 2 3 – 3 4 ∗ 4 5 + – – – –		
1 2 + 2 3 – 3 4 ∗ 4 5 + – – – – –		
1 2 + 3 4 – 5 ∗ 7 8 + 9 + + +		

11.4.

W poniższych wyrażeniach przyjmujemy, że $op_1, \ldots, op_{10}$ to znaki ze zbioru $\{+, -, *\}$. Podaj postać ONP poniższych wyrażeń.

X: (((((((((1 op_1 2) op_2 3) op_3 4) op_4 5) op_5 6) op_6 7) op_7 8) op_8 9) op_9 10)
ONP(X): .....
.....
Y: (1 op_1 (2 op_2 (3 op_3 (4 op_4 (5 op_5 (6 op_6 (7 op_7 (8 op_8 (9 op_9 10)))))))))
ONP(Y): .....
.....

Zadanie 12.
Wiązka zadań *Szyfr Beauforta*

Każdej z wielkich liter angielskiego alfabetu przyporządkowany jest kod ASCII, będący ośmiobitową liczbą naturalną. Poniższa tabela przedstawia kody ASCII dla liter od A do Z:

65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Dany jest następujący algorytm szyfrujący:

Dane:

n — liczba naturalna, wielkość tablicy *Tekst*[]

Tekst[] — tablica zawierająca kody ASCII liter tekstu do zaszyfrowania, elementy tablicy są numerowane od 0 do n-1

k — nieujemna liczba całkowita

Wynik:

Szyfrogram[] — tablica o długości *n* zawierająca kody ASCII liter zaszyfrowanego tekstu

```
     i ← 0
     dopóki (i<n) wykonuj
(*)      Szyfrogram[i] ← (90-Tekst[i]+k) mod 26 + 65
         i ← i+1
         k ← k+i
```

12.1.

Uzupełnij poniższą tabelę wartościami zmiennych *i* oraz *k* w wierszu oznaczonym (*) w kolejnych iteracjach algorytmu dla *n* równego 7:

Lp	*i*	*k*
1	0	20
2		
3		
4		
5		
6		

12.2.

Ile dla przedstawionego algorytmu jest możliwych początkowych wartości *k*, dających w efekcie różne szyfrogramy? Odpowiedź krótko uzasadnij.

12.3.

Uzupełnij poniższą tabelę. W wierszu pierwszym zaszyfruj przedstawionym w treści zadania algorytmem podany tekst jawny, w wierszu drugim odszyfruj szyfrogram zakodowany tym algorytmem.

k	*Tekst*[]	*Szyfrogram*[]
1	MAPA	
14		DAN

12.4.

Używając pseudokodu lub wybranego języka programowania, napisz algorytm dekodujący szyfrogram.

Dane:

n — liczba naturalna, wielkość tablicy *Tekst*[]

Szyfrogram[] — tablica o długości *n* zawierająca kody ASCII liter zaszyfrowanego tekstu

k — nieujemna liczba całkowita

Wynik:

Tekst[] — tablica zawierająca kody ASCII liter odszyfrowanego tekstu, elementy tablicy są numerowane od 0 do n-1

Zadanie 13.

Wiązka zadań *Zbiór punktów*

Na płaszczyźnie kartezjańskiej danych jest *n* punktów: P_1, P_2,..., P_n oraz jeszcze jeden punkt A.

Współrzędne punktów zapisane są w tablicach *PX[1..n]*, *PY[1..n]* — współrzędne punktu P_i dane są jako para (*PX[i]*, *PY[i]*) dla *i* = 1, 2, ..., n. Współrzędne punktu A to (*ax*, *ay*). Przeanalizuj poniższy algorytm i wykonaj poniższe polecenia.

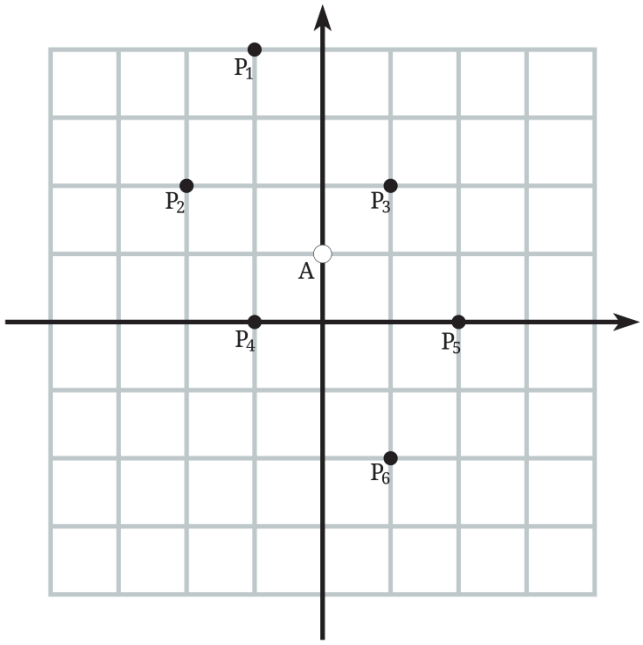
Dane:
PX[1..n], *PY[1..n]* — tablice liczb całkowitych zawierające współrzędne kolejnych punktów P_1, P_2,..., P_n
ax, *ay* — współrzędne punktu A.
Wynik:
odpowiedź TAK lub NIE.

```
funkcja sprawdz(PX[1..n], PY[1..n],ax,ay)
    dla i = 1, 2, ..., n wykonuj
        roznicax ← PX[i] – ax
        roznicay ← PY[i] – ay
        drugix ← ax – roznicax
```

```
        drugiy ← ay – roznicay
        znalezione ← fałsz
        dla k = 1, 2, ..., n wykonuj
                jeżeli drugix = PX[k] oraz drugiy = PY[k]
                        znalezione ← prawda
        jeżeli znalezione = fałsz
                wypisz NIE i zakończ wykonywanie algorytmu
wypisz TAK
```

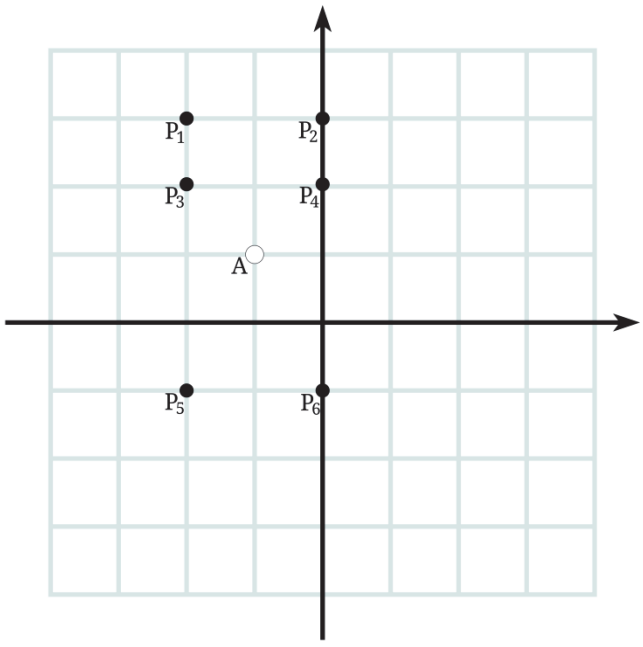
13.1.

Dla poniższych zbiorów punktów znajdź odpowiedź, jaką otrzymamy po wykonaniu powyższego algorytmu:



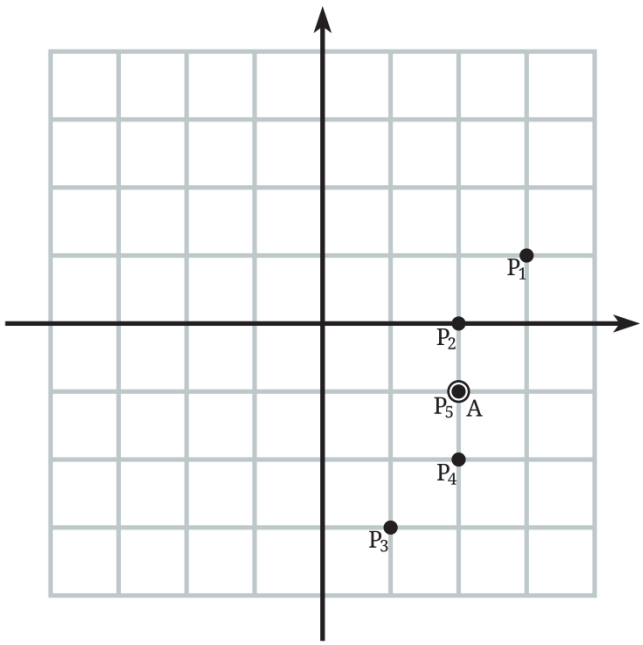
P_1 (-1,4)
P_2 (-2,2)
P_3 (1,2)
P_4 (-1,0)
P_5 (2,0)
P_6 (1,-2)
A (0, 1)

Odpowiedź algorytmu: .....



P_1 (-2,3)
P_2 (0,3)
P_3 (-2,2)
P_4 (0,2)
P_5 (-2,-1)
P_6 (0,-1)
A (-1,1)

Odpowiedź algorytmu: .....



P_1 (3,1)
P_2 (2,0)
P_3 (1,-3)
P_4 (2,-2)
P_5 (2,-1)
A (2,-1)

Odpowiedź algorytmu: .....

13.2.

Dane są punkty:

P_1 (3,–1)
P_2 (0,2)
P_3 (1,4)
P_4 (1,3)
P_5 (4,0)

oraz punkty P_6 i A o nieznanych współrzędnych. Wiadomo, że algorytm odpowiada TAK dla $n = 6$ oraz punktów P_1,..., P_6 i A. Podaj współrzędne punktów P_6 i A.

Zadanie 14.

Wiązka zadań *Działka*

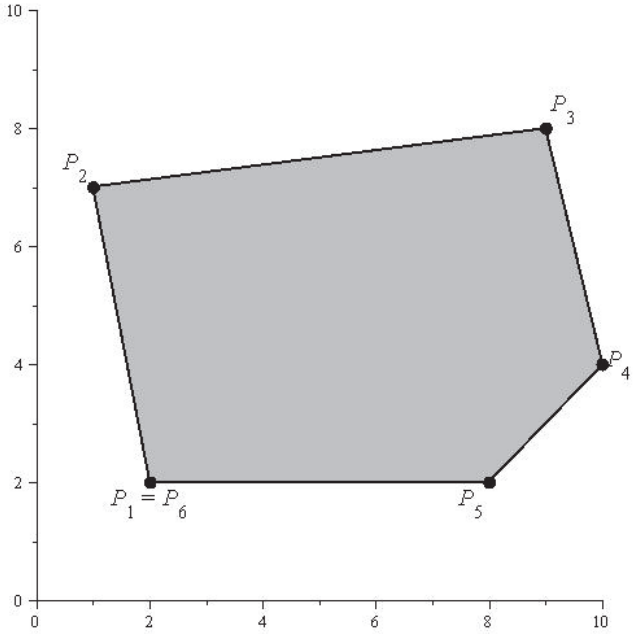
Pan G jest geodetą. Jego zadaniem jest wyznaczenie pola powierzchni działki budowlanej o kształcie pewnego wielokąta wypukłego. Aby wyznaczyć pole powierzchni działki, pan G obchodzi ją dookoła, tak aby działkę mieć zawsze po swojej prawej stronie. Podczas pomiaru zapisuje współrzędne kolejnych wierzchołków wielokąta, przy czym:

1. rozpoczyna obchód w punkcie $P_1 = [x_1, y_1]$;
2. współrzędne kolejno odwiedzonych wierzchołków oznacza:
$$P_2 = [x_2, y_2],\ P_3 = [x_3, y_3], \ldots,\ P_n = [x_n, y_n];$$
3. kończy obchód w punkcie początkowym, tzn. przyjmuje $P_{n+1} = P_1$.

Przykład

Poniższy obrazek przedstawia działkę ($n = 5$) oraz kolejne punkty pomiaru zanotowane przez geodetę:

$P_1 = [2,2]$, $P_2 = [1,7]$, $P_3 = [9,8]$, $P_4 = [10,4]$, $P_5 = [8,2]$, $P_6 = P_1$.



Następnie pan G wyznacza pole działki za pomocą wzoru

$$pole = \frac{1}{2}\sum_{i=1}^{n}(x_{i+1} - x_{i-1}) \cdot y_i,$$

gdzie x_0 przyjmuje jako x_n.

Obliczanie pola z podanego wzoru realizowane jest według następującego algorytmu:

Dane:

n — liczba całkowita, $n \geq 0$,

$x_1, y_1; x_2, y_2; \ldots; x_n, y_n$ — współrzędne punktów pomiaru.

Wynik:

pole działki

Algorytm:

$x_0 \leftarrow x_n$
$x_{n+1} \leftarrow x_1$
$y_{n+1} \leftarrow y_1$
$pole = 0$
dla $i = 1,2, \ldots, n$ **wykonuj**
    $pole \leftarrow pole + (x_{i+1} - x_{i-1}) \cdot y_i$
zwróć $\frac{pole}{2}$ **i zakończ**

14.1.

Przeanalizuj działanie powyższego algorytmu dla przykładu z rysunku i uzupełnij poniższą tabelkę.

i	x_i	y_i	$x_{i+1} - x_{i-1}$	$(x_{i+1} - x_{i-1}) \cdot y_i$
1	2	2		
2	1	7		
3	9	8		
4	10	4		
5	8	2		

Podaj obliczone pole działki: .....

14.2.

Niech punkty $A = [x_1, y_1],\ B = [x_2, y_2],\ C = [x_3, y_3]$ będą wierzchołkami pewnego trójkąta, podanymi w kolejności zgodnej z ruchem wskazówek zegara. Bazując na metodzie geodety, podaj wzór na pole trójkąta ABC.

14.3.

Wyraź wzorem liczbę operacji mnożenia oraz łączną liczbę operacji dodawania i odejmowania współrzędnych punktów, jaka zostanie wykonana przez podany algorytm podczas obliczania pola powierzchni działki o kształcie wielokąta n-wierzchołkowego.

Uzupełnij poniższą tabelkę:

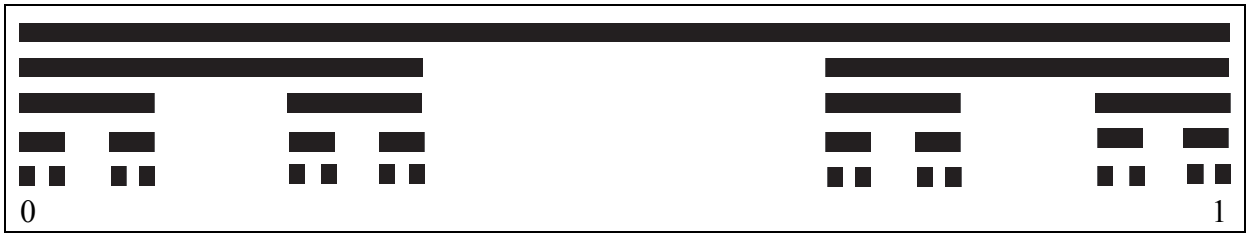
Działanie	**Liczba operacji**
dodawanie i odejmowanie	
mnożenie	

Zadanie 15.

Wiązka zadań *Zbiór Cantora*

Zbiór (fraktal) Cantora rzędu 0 jest równy odcinkowi jednostkowemu [0; 1]. Zbiór Cantora rzędu 1 uzyskujemy, dzieląc odcinek jednostkowy [0; 1] na trzy równe części i usuwając odcinek środkowy (pozostawiając zaś jego końce). Składa się on zatem z dwóch fragmentów odcinka jednostkowego: [0; 1/3], [2/3; 1].

Ogólnie zbiór Cantora rzędu *n*+1 tworzymy ze zbioru Cantora rzędu *n* w następujący sposób: każdy odcinek zbioru Cantora rzędu *n* dzielimy na trzy równe części i usuwamy odcinek środkowy, pozostawiając jego końce. Zgodnie z tą regułą zbiór Cantora rzędu 2 składa się z odcinków: [0; 1/9], [2/9; 3/9], [6/9; 7/9] i [8/9; 1].



Rysunek. Geometryczna ilustracja zbiorów Cantora rzędu 0, 1, 2, 3 i 4

15.1.

Uzupełnij poniższą tabelę, podając liczbę odcinków w zbiorach Cantora podanych rzędów.

n	Liczba odcinków w zbiorze Cantora rzędu *n*
0	1
1	2
2	4
3	8
5	
6	
9	
10	

Podaj ogólny wzór określający C(*n*), liczbę odcinków w zbiorze Cantora rzędu *n*:

C(*n*)=……..

15.2.

Zauważ, że każdy odcinek w zbiorze Cantora ustalonego rzędu ma tę samą długość. Uzupełnij poniższą tabelę, podając długość jednego odcinka w zbiorach Cantora podanych rzędów.

n	Długość jednego odcinka w zbiorze Cantora rzędu *n*
0	1
1	1/3
2	1/9
3	1/27
4	
5	
6	
7	

Podaj ogólny wzór określający długość D(*n*) jednego odcinka w zbiorze Cantora rzędu *n*:

D(*n*)=……..

15.3.

Uzupełnij poniższą listę odcinków zbioru Cantora rzędu 3, których końce zapisane są jako ułamki zwykłe **nieskracalne**:

[0; 1/27], [2/27; 1/9], ...
.....

Końce odcinków zbiorów Cantora można opisać w zwarty i regularny sposób w systemie trójkowym. W szczególności odcinki zbioru Cantora rzędu 1 zapisane w systemie trójkowym to

[0; 0,1], [0,2; 1],

natomiast odcinki zbioru Cantora rzędu 2 zapisane w systemie trójkowym to

[0; 0,01], [0,02; 0,1], [0,2; 0,21] i [0,22; 1].

Podaj poniżej odcinki zbiorów Cantora rzędu 3 zapisane w systemie o podstawie 3 (trójkowym):

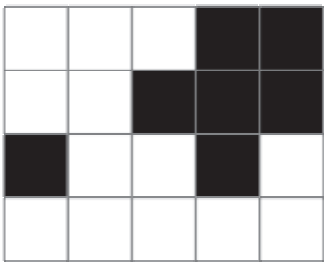
[0; 0,001], ...
.....

Zadanie 16.

Wiązka zadań *Odległość na planszy*

Dana jest prostokątna plansza o rozmiarach *m* na *n* (*m* wierszy, *n* kolumn), podzielona na *m·n* pól — jednostkowych kwadratów. Przez *pole (i, j)* rozumiemy pole, które jest w i-tym wierszu oraz j-tej kolumnie. Wiersze numerujemy kolejno od 1 do m, z góry do dołu, a kolumny kolejno od 1 do n,od lewej do prawej. **Dwa pola uważamy za *sąsiednie*, jeśli mają wspólny bok.**

Niektóre pola tej planszy zostały wycięte — na rysunku oznaczone są na czarno.



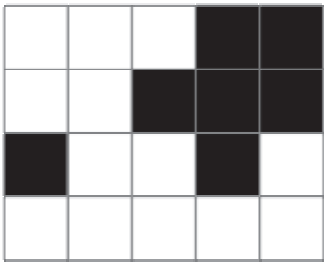
Poniżej podany jest algorytm, który tworzy tablicę *D[1..m, 1..n]* (o *m* wierszach oraz *n* kolumnach), odpowiadającą planszy, a następnie wpisuje liczby całkowite do komórek tablicy odpowiadających niewyciętym (białym) polom na planszy. Każda wpisana liczba mierzy w pewien sposób odległość odpowiedniego pola od lewego górnego rogu planszy (pola (1,1) — załóż, że to pole nigdy nie jest wycięte). Przeanalizuj algorytm i rozwiąż podanej niżej zadania.

```
dla każdego niewyciętego pola (i,j)
        D[i, j] ← -1

D[1, 1] ← 0
k ← 0
powtarzaj:
        jeśli nie ma takich pól (x,y), że D[x, y] = k
                zakończ algorytmu
        dla każdego pola (x,y) takiego, że D[x, y] = k wykonaj
                dla każdego pola (a,b) sąsiadującego z (x,y) wykonaj
                        jeśli (a,b) nie jest wycięte oraz D[a, b] = -1
                                D[a, b] ← k+1
        k ← k+1
```

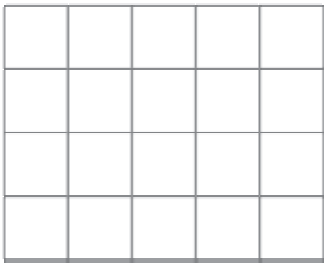
16.1.

Dla podanej poniżej planszy wyznacz wartości, które znajdą się w tablicy D po zakończeniu wykonywania algorytmu. Wpisz te wartości w odpowiednie pola planszy:



16.2.

Zaprojektuj taką planszę o rozmiarach *4·5*, dla której w wynikowej tablicy D pojawi się liczba 10. Odpowiedź podaj, kolorując na poniższym rysunku pewne pola na czarno oraz wpisując liczbę „10” w odpowiednim polu:



16.3.

Zaprojektuj taką planszę (wybranych przez siebie rozmiarów), dla której w wynikowej tablicy D po zakończeniu wykonywania algorytmu w pewnej komórce pozostanie liczba –1. Narysuj taką planszę i wpisz -1 na wszystkich białych polach, na których ta wartość pozostanie do końca działania algorytmu.

Zadanie 17.

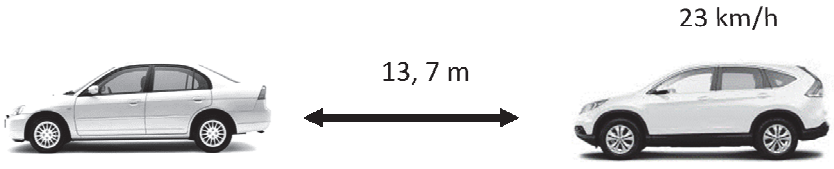
Wiązka zadań *System hamowania*

Nowoczesne samochody mają wbudowany komputer z systemem wspomagającym prowadzenie pojazdu. Jedną z jego funkcji jest aktywacja systemu hamowania w momencie, gdy samochód znajdzie się zbyt blisko pojazdu znajdującego się przed nim.

Automatyczne hamowanie wykorzystuje czujniki, które mierzą prędkość samochodu oraz jego odległość od pojazdu znajdującego się przed nim. Reakcja systemu zależy od prędkości samochodu, w którym jest on uruchomiony. W zadaniach ograniczamy się do analizy działania systemu przy małych prędkościach samochodu.

Przyjmujemy, że automatyczne uruchomienie hamowania następuje, jeśli odległość od najbliższego pojazdu jest **mniejsza niż** 15 metrów oraz prędkość samochodu, w którym działa system, wynosi **mniej niż** 30 km/h (zobacz poniższy rysunek). Po automatycznym uruchomieniu hamowania prędkość samochodu będzie spadać, a hamulce będą włączone do momentu osiągnięcia przez samochód prędkości 0 km/h, niezależnie od tego, jak będzie się zmieniać odległość od najbliższego pojazdu.

W analizie działania systemu hamowania przyjmuje się, że samochód nie przekracza prędkości 350 km/h.



Działanie systemu automatycznego hamowania zostało poniżej błędnie opisane w pseudokodzie.

odczytaj *prędkość samochodu*;
odczytaj *odległość od najbliższego pojazdu*;
jeżeli *prędkość samochodu* < 30 **lub** *odległość od najbliższego pojazdu* < 15:
    dopóki *prędkość samochodu* = 100 **wykonuj**
        wyślij *polecenie hamowania* do układu hamulcowego;
        odczytaj *prędkość samochodu*

17.1.

W powyższym pseudokodzie występują dwa błędy powodujące niezgodność z podanym wcześniej opisem. Znajdź i popraw oba błędy.

Błędny zapis w pseudokodzie	Poprawny zapis w pseudokodzie

17.2.

Przyjmujemy, że system hamowania odczytuje prędkość samochodu i odległość od najbliższego pojazdu na tyle często, że sąsiednie pomiary mogą się różnić co najwyżej o odpowiednio 1 km/h i 1 m (pomiary prędkości podawane są w km/h, w zaokrągleniu do wartości całkowitych; pomiary odległości podawane są w m, również w zaokrągleniu do wartości całkowitych). Oceń, czy podane w poszczególnych wierszach poniższej tabeli dane testowe są danymi standardowym (niezmieniające stanu sytemu hamowania), brzegowymi (wartości, które mogą powodować włączenie lub wyłączenie systemu hamowania) czy też danymi niezgodnymi ze specyfikacją.

Dane testowe	Typ danych testowych
prędkość samochodu — 29 km/h, ***odległość między samochodami — 1 m***	**brzegowe**
prędkość samochodu — 400 km/h, *odległość od najbliższego pojazdu — 0 m*	
prędkość samochodu — 13 km/h, *odległość od najbliższego pojazdu — 8 m*	
prędkość samochodu — 45 km/h, *odległość od najbliższego pojazdu — 17 m*	
prędkość samochodu — 0 km/h, *odległość od najbliższego pojazdu — 17 m*	

17.3.

Wskaż, które z poniższych dokończeń zdań są prawdziwe (P), a które fałszywe (F), wstawiając znak X w odpowiedniej kolumnie.

Celem testowania systemu dla różnych typów danych jest

	P	F
upewnienie się, że system poradzi sobie z odczytem parametrów z czujników i tempo jego reakcji pozwoli uniknąć zderzenia.		
zwiększenie precyzji pomiarów prędkości i odległości.		
zmniejszenie długości kodu programu.		
sprawdzenie, czy system działa zgodnie z podaną specyfikacją.		

17.4.

Uzupełnij poniższą tabelę, wpisując w ostatniej kolumnie słowo *włączony, wyłączony* (gdy automatyczne hamowanie będzie przy danych odczytach włączone/wyłączone niezależnie od wartości poprzedniego pomiaru) lub *nieustalony* (gdy aktywność automatycznego hamowania zależy zarówno od bieżącego, jak i od poprzedniego pomiaru).

Prędkość samochodu	**Odległość między samochodami**	**Stan automatycznego hamowania**
poniżej 30 km/h	poniżej 15m	włączony
poniżej 30 km/h	równa15m	
równa 30 km/h	poniżej 15m	
równa 30 km/h	równa15m	
powyżej 30 km/h	dowolna	

1.2. Tworzenie algorytmów

Zadanie 18.

Wiązka zadań *Największy wspólny dzielnik*

Opisana poniżej funkcja wyznacza największy wspólny dzielnik (NWD) dwóch liczb całkowitych dodatnich.

Specyfikacja

Dane**:**

liczby całkowite dodatnie a i b.

Wynik**:**

Największy wspólny dzielnik liczb a i b.

funkcja *NWD* (*a*, *b*)
    dopóki $b \neq 0$ **wykonuj**
        $r \leftarrow a \bmod b$ (*)
        $a \leftarrow b$
        $b \leftarrow r$
    zwróć *a* **i zakończ**

18.1.

Przeanalizuj działanie funkcji *NWD* i dla wskazanych argumentów podaj liczbę wykonań operacji modulo:

a	*b*	liczba operacji mod (*)
25	15	
116	324	
762	282	

18.2.

Wykorzystując funkcję *NWD* oraz następującą zależność:

$$\text{NWD}(a_1, a_2, \dots, a_n) = \text{NWD}\,(\text{NWD}\,(a_1, a_2, \dots, a_{n-1}), a_n),$$

możemy wyznaczyć największy wspólny dzielnik *n* liczb całkowitych dodatnich $a_1, a_2, \dots, a_n$.

Przykład

$$\text{NWD}(15, 24, 60) = \text{NWD}\,(\text{NWD}\,(15,24), 60) = \text{NWD}\,(3,60) = 3.$$

Uzupełnij poniższą tabelkę i dla wskazanych *n* liczb całkowitych dodatnich $a_1, a_2, \dots, a_n$ oblicz ich największy wspólny dzielnik.

$a_1, a_2, \dots, a_n$	$\text{NWD}(a_1, a_2, \dots, a_n)$
36, 24, 72, 150	
119, 187, 323, 527, 731	
121, 330, 990, 1331, 110, 225	

18.3.

Napisz algorytm, który korzystając z funkcji *NWD* (*a*, *b*), wyznaczy największy wspólny dzielnik ciągu liczb $a_1, a_2, \dots, a_n$.

Dane:

$a_1, a_2, \dots, a_n$ — liczby całkowite dodatnie.

Wynik:

Największy wspólny dzielnik liczb $a_1, a_2, \dots, a_n$.

Zadanie 19.

Wiązka zadań *Zakupy międzyplanetarne*

Mieszkańcy galaktyki Różnoliczbowo zamieszkują 9 planet: Liczbowo$_2$, Liczbowo$_3$, …, Liczbowo$_{10}$. Na każdej planecie Liczbowo$_i$ jej mieszkańcy posługują się systemem liczbowym o podstawie *i*. Na każdej planecie wszystkie ceny są liczbami naturalnymi.

19.1.

Mieszkańcy czterech sąsiadujących planet: Liczbowo$_2$, Liczbowo$_4$, Liczbowo$_8$ oraz Liczbowo$_{10}$ często podróżują pomiędzy tymi planetami i kupują różne towary. W poniższej tabeli znajdują się ceny wybranych towarów zakupionych przez jedną osobę na różnych planetach. Uzupełnij tabelę, przeliczając **podane** ceny na systemy liczbowe wszystkich czterech planet.

Towar	**Cena towaru zapisana w systemie liczbowym planety**			
	Liczbowo$_2$	**Liczbowo$_4$**	**Liczbowo$_8$**	**Liczbowo$_{10}$**
Kozaki	10111011			
Płaszcz			724	
Skuter				1458

19.2.

Na różnych planetach ten sam towar może mieć różną cenę, na przykład cena ciasta kokosowego na planecie Liczbowo$_{10}$ wynosi 38_{10}, zaś na planecie Liczbowo$_8$ jego cena wynosi 55_8 (równą 45_{10}).

Mieszkańcy planety Liczbowo$_{10}$ są bardzo oszczędni i przed zakupami porównują ceny towarów na wybranych planetach. Uzupełnij w poniższej tabeli relacje (>, <, =) pomiędzy poszczególnymi cenami.

Liczbowo$_x$	Relacja	Liczbowo$_y$
110000100_2	>	556_8
3123_4		1747_8
110_{10}		11010_3
266_9		110100_3
110111101_2		674_8

19.3.

Sprzedawcy sklepów planety Liczbowo$_i$ wyliczają wartość zakupów klientów, sumując ceny zakupionych towarów w systemie obowiązującym na ich planecie; stosują przy tym metodę dodawania pisemnego.

Przykłady

Dodawanie w systemie o podstawie 2:

$$\begin{array}{cccccccc} & & & 1 & 0 & 1 & 0 \\ & & 1 & 0 & 0 & 1 & 1 \\ & & & 1 & 1 & 0 & 1 \\ +_2 & & 1 & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Dodawanie w systemie o podstawie 4:

$$\begin{array}{cccc} & & 2 & 2 \\ & 1 & 0 & 3 \\ & & 3 & 1 \\ +_4 & 1 & 2 & 1 \\ \hline 1 & 0 & 0 & 3 \end{array}$$

Podsumuj rachunki pana Dwójkowskiego (z planety Liczbowo$_2$) oraz pana Czwórkowskiego (z planety Liczbowo$_4$).

Rachunek pana Dwójkowskiego:

$$\begin{array}{cccccccc} & & & & 1 & 0 & 1 & 1 \\ & & & 1 & 0 & 0 & 1 & 0 \\ & & 1 & 1 & 0 & 1 & 1 & 1 \\ & & & 1 & 1 & 0 & 1 & 0 \\ & & & 1 & 1 & 0 & 0 & 1 \\ +_2 & & 1 & 0 & 1 & 0 & 1 & 1 \\ \hline \text{SUMA} & & \ldots\ldots & & & & & \end{array}$$

Rachunek pana Czwórkowskiego:

			3	3
		1	0	2
		3	1	1
		1	3	1
		1	2	3
	$+_4$	3	0	1
	SUMA	…….		

Podaj różnicę wartości obu rachunków w systemie obowiązującym na planecie Liczbowo_{10}.

Różnica rachunków w systemie obowiązującym na planecie Liczbowo_{10} wynosi: ……..

19.4.

Właściciele sieci sklepów ulokowanych na wszystkich planetach galaktyki postanowili dostarczyć do sklepów kalkulatory, które będą dodawały ceny w opisany powyżej sposób. Aby pomóc właścicielom, podaj algorytm dodawania (w postaci pseudokodu lub w języku programowania), który dla dwóch liczb a i b zapisanych w systemie o podstawie $p \in [2,9]$ wyznacza i wypisuje wartość sumy $a +_p b$ zapisaną w systemie o podstawie p zgodnie z poniższą specyfikacją. Twój algorytm **nie może** dokonywać zamiany liczb a i b na inny system pozycyjny.

Specyfikacja

Dane:

p — podstawa systemu, liczba naturalna z przedziału [2,9],

n — liczba cyfr liczb naturalnych a, $b \leq 255$ (przyjmujemy, że krótsza liczba jest uzupełniona z lewej strony zerami, tak aby obie liczby miały taką samą długość),

A[n], A[n−1],..., A[1] – kolejne cyfry liczby a zapisanej w systemie o podstawie p,

B[n], B[n−1],..., B[1] – kolejne cyfry liczby b zapisanej w systemie o podstawie p,

Wyniki:

wartość liczby $c = a +_p b$ zapisana w systemie o podstawie p w postaci ciągu cyfr C[n+1], C[n],…,C[1].

Przykład

Dla następujących danych:

$p = 4$

$n = 4$,

Liczba $a = 3122_4$

Liczba $b = 21_4$

Wynikiem jest liczba $c = 3203_4$

Zawartość tablic A, B, C:

i	5	4	3	2	1
A[i]		3	1	2	2
B[i]		0	0	2	1
C[i]	0	3	2	0	3

Zadanie 20.
Wiązka zadań *Liczba narcystyczna*

Niech dana będzie liczba naturalna x, której zapis dziesiętny ma n cyfr:

$$x = a_{n-1}10^{n-1} + a_{(n-2)}10^{n-2} + \cdots + a_1 10 + a_0 \qquad (a_{n-1} \neq 0).$$

Powiemy, że liczba x jest *narcystyczna*, jeśli suma jej cyfr podniesionych do potęgi n-tej jest równa x, tzn.

$$a_{n-1}^n + a_{n-2}^n + \cdots + a_1^n + a_0^n = x.$$

Na przykład liczba 1634 jest narcystyczna, ponieważ

$$1^4 + 6^4 + 3^4 + 4^4 = 1634.$$

Powiemy, że liczba x jest *B-narcystyczna*, jeśli jej zapis w systemie o podstawie B ma n cyfr, których suma n-tych potęg jest równa x, tzn.

$$x = a_{n-1}B^{n-1} + a_{(n-2)}B^{n-2} + \cdots + a_1 B + a_0 \quad \text{oraz} \quad a_{n-1}^n + a_{n-2}^n + \cdots + a_1^n + a_0^n = x.$$

Na przykład liczba 289 jest 5-narcystyczna, ponieważ

$$289 = (2124)_5 = 2 \cdot 5^3 + 1 \cdot 5^2 + 2 \cdot 5 + 4 \text{ oraz } 289 = 2^4 + 1^4 + 2^4 + 4^4.$$

20.1.

Uzupełnij brakujące cyfry tak, aby powstałe liczby były liczbami narcystycznymi.

3□1

40□

54□48

20.2.

Sprawdź, które z poniższych liczb (podanych w systemie dziesiętnym) są B-narcystyczne dla podanych wartości B. Uzupełnij poniższą tabelkę, wpisując *prawda* lub *fałsz* w zależności od tego, czy liczba jest B-narcystyczna, czy też nie.

x	B	*prawda/fałsz*
3433	6	
4890	5	
8956	3	
15345	2	

20.3.

Napisz algorytm (w pseudokodzie lub wybranym języku programowania), który sprawdza, czy dana liczba x jest B-narcystyczna.

Dane:

x — liczba całkowita, $x \geq 0$,

B — liczba całkowita, $B \geq 2$.

Wynik:

TAK, jeśli liczba x jest B-narcystyczna, NIE — w przeciwnym przypadku.

Zadanie 21.

Wiązka zadań *Szybkie podnoszenie do potęgi*

W algorytmach szybkiego potęgowania można wykorzystać binarną reprezentację wykładnika dla obliczenia wartości x^k, gdzie k jest liczbą naturalną, $k \neq 0$, zaś x jest liczbą rzeczywistą. Przyjmijmy, że binarnym rozwinięciem wykładnika k jest ciąg $(k_n k_{n-1} k_{n-2} \ldots k_2 k_1 k_0)_2$.

Jedna z metod wyznaczania x^k polega na obliczaniu potęg liczby x dla wykładników o binarnych reprezentacjach:

k_n,

$k_n k_{n-1}$,

$k_n k_{n-1} k_{n-2}$,

…,

$k_n k_{n-1} k_{n-2} \ldots k_1$,

$k_n k_{n-1} k_{n-2} \ldots k_1 k_0$,

Inaczej mówiąc, uwzględniamy coraz dłuższe fragmenty ciągu $(k_n k_{n-1} k_{n-2} \ldots k_2 k_1 k_0)_2$. W pierwszym kroku przyjmujemy, że wynik jest równy x, gdyż $k_n = 1$. Znając wartość x do potęgi o binarnym zapisie $(k_n k_{n-1} k_{n-2} \ldots k_i)_2$, możemy łatwo wyliczyć x do potęgi o binarnym zapisie $(k_n k_{n-1} k_{n-2} \ldots k_i k_{i-1})_2$: podnosimy dotychczasowy wynik do kwadratu (do czego wystarczy jedno mnożenie). Jeśli $k_{i-1} = 1$, dodatkowo mnożymy uzyskany wynik przez x.

Przykład

Niech $k = 13 = (1101)_2$. Kolejne wyznaczane w naszym algorytmie potęgi to:

$$x^1,\ x^3 = x^2 x,\ x^6 = (x^3)^2,\ x^{13} = (x^6)^2 x,$$

zaś liczba wykonanych mnożeń jest równa 5 (zauważ, że aby obliczyć x^3, musisz najpierw obliczyć x^2, a aby obliczyć x^2, musisz wykonać jedno mnożenie: $x^2 = x \cdot x$).

21.1.

Korzystając z przedstawienia wykładnika w postaci binarnej, podaj kolejne potęgi liczby x wyznaczane powyższą metodą przy obliczaniu x^{38}.

21.2.

Uzupełnij tabelkę. Oblicz, ile mnożeń wykonywanych jest dla kolejnych wykładników.

k	**reprezentacja binarna *k***	**liczba mnożeń**
4	100	2
5	101	3
6		
7		
8		
15		
16		
22		
32		

21.3.

W wybranej przez siebie notacji (lista kroków, pseudokod, język programowania) napisz algorytm, który dla zadanej binarnej reprezentacji liczby naturalnej k, $k \neq 0$, oraz rzeczywistej liczby x oblicza wartość x^k zgodnie z metodą opisaną na początku zadania.

Specyfikacja

Dane:

x — liczba rzeczywista,

n — liczba całkowita nieujemna,

$k_n k_{n-1} k_{n-2} \ldots k_1 k_0$ — ciąg tworzący binarną reprezentację wykładnika k,

Wynik:

liczba rzeczywista $p = x^k = \underbrace{x \cdot x \cdot \ldots \cdot x}_{k\ razy}$

Zadanie 22.

Wiązka zadań *Schemat Hornera*

Schemat Hornera jest bardzo efektywną metodą obliczania wartości wielomianu

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0,$$

gdzie dane liczby rzeczywiste $a_0, a_1, \ldots, a_n$ nazywamy *współczynnikami*, a liczba całkowita $n \geq 0$ oznacza *stopień* wielomianu.

Schemat bazuje na zależności

$$P(x) = x(a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_2 x^1 + a_1) + a_0 = x \cdot Q(x) + a_0,$$

gdzie

$$Q(x) = a_n x^{n-1} + a_{n-1}x^{n-2} + \cdots + a_2 x^1 + a_1.$$

Stąd otrzymujemy następujący schemat obliczania wartości $P(x)$:

Dane:

n — liczba całkowita, $n \geq 0$,

x — liczba rzeczywista,

$a_0, a_1, \ldots, a_n$ — liczby rzeczywiste.

Wynik:

wartość $P(x)$

Algorytm (schemat Hornera):

$w \leftarrow a_n$
dla $k = n-1, n-2, \ldots, 0$ **wykonuj**
(*) $\quad w \leftarrow x \cdot w + a_k$
zwróć w **i zakończ**

22.1.

Uzupełnij poniższą tabelkę, podając wartości danych, jakie należy przyjąć w powyższym schemacie, aby wyznaczyć wartość $P(6)$ dla wielomianu

$$P(x) = 10x^5 - 13x^4 + x^3 + 2x^2 - 8x + 7.$$

Dane	Wartości
liczba naturalna n	
liczba rzeczywista x	
liczby rzeczywiste $a_0, a_1, \ldots, a_n$	

22.2.

Uzupełnij poniższą tabelkę, wyrażając wzorem liczbę operacji mnożenia i dodawania, jaka zostanie wykonana przez schemat Hornera (w wierszu oznaczonej przez (*)) dla danego wielomianu

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0.$$

Działanie	Liczba operacji
Dodawanie	
Mnożenie	

22.3.

Przeanalizuj działanie schematu Hornera podczas obliczania wartości $P(2)$ dla wielomianu

$$P(x) = 4x^6 - 3x^5 + 2x^3 - 5x^2 + 7x + 9.$$

W poniższej tabeli wpisz wartości w obliczane przez algorytm w linii (*)

	Wartość w
$k = 5$	
$k = 4$	
$k = 3$	
$k = 2$	
$k = 1$	
$k = 0$	

Podaj wynik, jaki zwróci algorytm: .....

22.4.

Wielomianem parzystym nazywamy wielomian stopnia $2n$ postaci

$$R(x) = a_n x^{2n} + a_{n-1} x^{2n-2} + \cdots + a_2 x^4 + a_1 x^2 + a_0,$$

tzn. taki, w którym występują tylko parzyste potęgi zmiennej x.

Bazując na schemacie Hornera, napisz algorytm o poniższej specyfikacji (w pseudokodzie lub wybranym języku programowania), który oblicza wartość parzystego wielomianu $R(x)$.

Dane:
n — liczba całkowita, $n \geq 0$,
x — liczba rzeczywista,
$a_0, a_1, \ldots, a_n$ — liczby rzeczywiste.
Wynik: wartość $R(x)$.

Przy ocenie rozwiązania będzie brana pod uwagę liczba operacji mnożenia i dodawania wykonywanych przez algorytm.

Zadanie 23.

Wiązka zadań *Obliczanie całkowitego pierwiastka kwadratowego*

Całkowity pierwiastek kwadratowy z liczby naturalnej x jest największą liczbą naturalną p, która spełnia nierówność $p^2 \leq x$. Poniższy algorytm służy do obliczania tej wartości przybliżonej.

Specyfikacja

Dane:

x — liczba naturalna

Wynik:

p — liczba naturalna spełniająca warunek $p^2 \leq x$ i $(p+1)^2 > x$

Algorytm

$i \leftarrow 0$
$a \leftarrow 0$
$r \leftarrow 1$
dopóki $a \leq x$ **wykonuj**
    $i \leftarrow i + 1$
    $a \leftarrow a + r$
    $r \leftarrow r + 2$
zwróć $i - 1$ **i zakończ**

23.1.

Niech $x = 21$. Przeanalizuj działanie powyższego algorytmu i uzupełnij wartości zmiennych a i r dla kolejnych wartości i podanych w tabeli.

Wartość i	Wartość a	Wartość r
0		
1		
2		
3		
4		
5		

23.2.

Zdecyduj, które z poniższych zdań są w odniesieniu do opisanego algorytmu prawdziwe (**P**), a które fałszywe (**F**). **Zaznacz znakiem X** odpowiednią rubrykę w tabeli .

	P	F
Konstruuje ciąg kwadratów kolejnych liczb naturalnych.		
Znajduje dokładną wartość pierwiastka z liczby x.		
Oblicza kolejne nieparzyste liczby naturalne.		
Wykonuje dokładnie tyle iteracji pętli, ile wynosi pierwiastek całkowity z liczby x.		

23.3.

Napisz algorytm obliczania całkowitego pierwiastka z liczby naturalnej, który wykorzystuje następującą zależność rekurencyjną definiującą ciąg x_n zbieżny do pierwiastka kwadratowego z liczby x:

$$\begin{cases} x_0 = \frac{x}{2} \\ x_{n+1} = \frac{1}{2}\left(x_n + \frac{x}{x_n}\right) \end{cases}$$

Specyfikacja
Dane:

x — liczba naturalna

Wynik:

p — liczba naturalna spełniająca warunek $p^2 \leq x$ i $(p+1)^2 > x$

Zadanie 24.
Wiązka zadań *Poszukiwania*

Dana jest liczba naturalna $n > 0$ oraz uporządkowana tablica liczb całkowitych $T[1..n]$. Rozważmy następującą funkcję F dla trzech argumentów p, k, e, które są liczbami całkowitymi dodatnimi:

```
funkcja F (p, k, e)
    jeżeli (k = p)
        jeżeli (T [p] > e)
            zwróć p i zakończ
        w przeciwnym razie
            zwróć p + 1  i zakończ
    w przeciwnym razie
        s ← (p+k) div 2
        jeżeli T [s] > e
            zwróć F (p, s, e)
        w przeciwnym razie
            zwróć F (s+1, k,  e)
```

24.1.

Przeanalizuj działanie funkcji F i uzupełnij poniższą tabelkę dla $p = 1$, $k = 5$, $e = 10$:

T	F (p, k, e)
[3, 4, 6, 8, 9]	
[15, 16, 18, 22, 24]	
[2, 10, 16, 24, 26]	
[1, 3, 10, 10, 18]	

24.2.

Zdecyduj, które z dokończeń podanego niżej zdania czynią z niego zdanie prawdziwe. Zaznacz to **znakiem X** w odpowiednich miejscach tabeli.

Funkcja F wykorzystuje

metodę zachłanną.	
strategię „dziel i zwyciężaj”.	
programowanie dynamiczne.	

24.3.

Zdecyduj, które z dokończeń podanego niżej zdania czynią z niego zdanie prawdziwe. Zaznacz to **znakiem X** w odpowiednich miejscach tabeli.

Liczba wywołań rekurencyjnych funkcji F dla początkowych wartości $p = 1$ oraz $k = n$, będącej potęgą dwójki, jest w najgorszym przypadku równa

n div 2.	
$\sqrt{n}$.	
$\log_2 n$.	

24.4.

Napisz algorytm, który korzystając z funkcji F, obliczy, ile jest liczb należących do przedziału [*a*, *b*] w uporządkowanej niemalejąco tablicy *T* [1..*n*], składającej się z liczb całkowitych dodatnich.

Specyfikacja

Dane**:**

n — liczba całkowita dodatnia

T [1..*n*] — uporządkowana tablica *n* liczb całkowitych, złożona z różnych liczb całkowitych dodatnich, taka że $T[1] \leq T[2] \leq \ldots \leq T[n]$

a — liczba całkowita dodatnia

b — liczba całkowita dodatnia

Wynik:

w — liczba elementów tablicy *T* [1..*n*] należących do przedziału [*a*, *b*].

Zadanie 25.

Wiązka zadań *Palindromy*

Palindrom to słowo lub zdanie, które czytane od lewej do prawej i od prawej do lewej brzmi tak samo. W zdaniu, które jest palindromem, ignorujemy odstępy między słowami. Ważne jest tylko, że kolejność liter przy czytaniu od lewej do prawej i od prawej do lewej jest taka sama. Dla potrzeb zadania przyjmujemy, że używamy tylko małych liter alfabetu łacińskiego. Poniżej podajemy przykłady słów i zdań, które są palindromami:

```
kajak
anna
elf ukladal kufle
trafili popili fart
ewo zeby tu byly buty bezowe
```

Poniższy algorytm sprawdza, czy podane na wejściu słowo jest palindromem**.**

Specyfikacja

Dane:

S[1..*d*] — słowo zapisane w tablicy znaków, gdzie *d*>1 oznacza liczbę znaków w słowie

Wynik:

TAK — gdy słowo *S* jest palindromem, NIE — w przeciwnym przypadku

```
    Algorytm:
     d ← długość(S)
     i ← d div 2
(*)  dopóki (i > 0) i (S[i] = S[d – i+1]) wykonuj
          i ← i – 1
     jeżeli i = 0
               zwróć TAK i zakończ,
     w przeciwnym razie
               zwróć NIE i zakończ
```

Uwaga: **div** oznacza operator dzielenia całkowitego, a wartością funkcji *długość* jest liczba znaków w słowie.

25.1.

Podaj przykład słowa o długości 9, niebędącego palindromem, dla którego powyższy algorytm wykonuje największą możliwą liczbę powtórzeń w pętli oznaczonej (*).

25.2.

Napisz algorytm, który sprawdza, czy zdanie jest palindromem.

Specyfikacja
Dane:

Zdanie[1..*d*] — zdanie o długości *d* znaków zapisane w tablicy znaków, składające się z małych liter alfabetu łacińskiego i spacji, w tym co najmniej jednej litery.

Wynik:

TAK, gdy *Zdanie* jest palindromem, NIE — w przeciwnym razie

Zadanie 26.

Wiązka zadań *Podobieństwo słów*

Słowo *X* nazywać będziemy *k–podrzędnym* względem słowa *Y*, jeśli **wszystkie litery występujące w *X* występują również w *Y*** oraz w słowie *Y* występuje dokładnie ***k* różnych liter**, które **nie** występują w słowie *X*.

Słowo *X* jest podrzędne względem słowa *Y*, gdy *X* jest *k*-podrzędne względem *Y* dla jakiegoś $k \geq 0$.

Przykład

Słowo *X=ABCAB* jest 1-podrzędne względem słowa *Y=BAACD* (w słowie *X* występują litery A, B, C; w słowie *Y* występują litery A, B, C, D). Podobnie słowo *X=ABCAB* jest 0-podrzędne względem *BAC* oraz nie jest podrzędne względem słów *ABDAB* i *ABAB* (litera *C* występuje w słowie X, a nie występuje w słowach *ABDAB* i *ABAB*). Zamieniając słowa rolami, możemy stwierdzić, że słowo *ABDAB* nie jest podrzędne względem słowa *ABCAB*, a słowo *ABAB* jest 1-podrzędne względem *ABCAB*.

Uwaga

W poniższych zadaniach przyjmujemy, że w słowach mogą występować tylko litery *A, B, C, D, E, F, G, H, I, J*. Ponadto w algorytmach dostępna jest funkcja *dlugosc*, która zwraca długość słowa będącego jej argumentem, oraz funkcja *kod* o wartościach podanych w poniższej tabeli:

litera	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
kod(litera)	1	2	3	4	5	6	7	8	9	10

26.1.

Uzupełnij poniższą tabelę, wpisując w kolumnie Podrzędność słowo NIE, jeśli słowo *X* nie jest podrzędne względem słowa *Y*, a w przeciwnym wypadku — liczbę *k* taką, że *X* jest *k*-podrzędne względem *Y*.

Słowo *X*	Słowo *Y*	Podrzędność
HHGGFFEEDDCCBBAA	*ABCDEFGH*	
DCBADCBA	*FGHABCJD*	
ABCDE	*ABCCBAE*	
AAAAA	*AA*	
ABA	*ACA*	
ACEGJ	*ABCDEFGHJ*	

26.2.

Dany jest następujący **algorytm A**:

dla *i*=1,2,…,10 **wykonuj**
    Czyjest[*i*] ← fałsz
d ← *dlugosc*(*Y*)
dla *i*=1,2,…,*d* **wykonuj**
    lit ← *Y*[*i*]
    Czyjest[kod(*lit*)] ← prawda
d ← *dlugosc*(*X*)
czyp ← prawda
dla *i*=1,2,…,*d* **wykonuj**
    lit ← *X*[*i*]
    czyp ← *czyp* **i** *Czyjest*[kod(*lit*)]
jeżeli *czyp*=prawda
    zwróć 1
w przeciwnym razie
    zwróć 0

Podaj wynik działania powyższego algorytmu dla wartości *X* i *Y* z poniższej tabeli:

X	*Y*	wynik algorytmu A
HHGGFFEEDDCCBBAA	*ABCDEFGH*	
DCBADCBA	*FGHABCJD*	
ABCDE	*ABCCBA*	
AAAAA	*AA*	
AA	*AAAAA*	
ACEGJ	*ABCDEFGH*	

Uzupełnij podaną poniżej specyfikację algorytmu A.

Specyfikacja

Dane: X, Y — słowa, w których występują tylko litery ze zbioru {*A, B, C, D, E, F, G, H, I, J*}

Wynik: .....
.....

26.3.

Uzupełnij brakujące fragmenty poniższego algorytmu B, tak aby realizował on podaną specyfikację.

Specyfikacja

Dane:

X, Y — słowa, w których występują tylko litery ze zbioru {*A,B,C,D,E,F,G,H,I,J*}

Wynik:

k — liczba całkowita, taka że

— *X* jest *k*-podrzędne względem *Y*, jeśli $k \geq 0$,

— *X* nie jest podrzędne względem *Y*, gdy $k = -1$.

Algorytm B:

```
dla i=1,2,…,10 wykonuj
        Czy_x[i] ← fałsz
        Czy_y[i] ← fałsz
dx ← dlugosc(X)
dla i=1,2,…,dx wykonuj
        lit ← X[i]
        Czy_x[kod(lit)] ← prawda
dy ← dlugosc(Y)
dla i=1,2,…,dy wykonuj
        lit ← Y[i]
        Czy_y[kod(lit)] ← prawda
k ← 0
dla i=1,2,…,10 wykonuj
        jeżeli Czy_y[i]=prawda oraz Czy_x[i]= .....
                k← .....
        jeżeli Czy_y[i]=..... oraz Czy_x[i]=prawda
                zwróć −1 i zakończ
zwróć k
```

26.4.

Słowa *X* i *Y* nazywać będziemy *równoważnymi*, jeśli każda litera występuje tyle samo razy w słowie *X* i w słowie *Y*.

Przykład

Słowa *ABCA* i *BCAA* są równoważne, natomiast *ABCA* nie jest równoważne ze słowami *ABCC*, *BABB* i *ABCAD*.

Podaj algorytm, który sprawdza, czy dwa podane słowa są równoważne. Twój algorytm powinien realizować następującą specyfikację:

Specyfikacja

Dane:

X, *Y* — słowa, w których występują tylko litery ze zbioru {*A, B, C, D, E, F, G, H, I, J*}

Wynik:

1 — gdy *X* i *Y* są równoważne, 0 — w przeciwnym razie.

Zadanie 27.

Wiązka zadań *Dopasowanie z błędem*

W podanym *tekście*, złożonym z małych liter alfabetu łacińskiego, wyszukujemy słowo zwane *wzorcem*. Celem jest znalezienie takiego fragmentu tekstu, który jest albo dokładnie równy wzorcowi, albo jest mu równy z jednym błędem, czyli różni się od niego najwyżej jedną literą.

Na przykład wzorzec *para* można znaleźć w tekście *parawan* albo *aparat*, a z jednym błędem — w tekście *opera* albo *spadanie*. Będziemy zakładać, że tekst jest co najmniej tak samo długi jak wzorzec, a wzorzec składa się z co najmniej dwóch liter.

27.1.

Dla podanych wzorców i tekstów podaj, czy wzorzec występuje w tekście dokładnie, z jednym błędem, czy też w ogóle w nim nie występuje. Do tabeli wpisz odpowiednio „dokładnie", „z błędem" lub „nie".

Wzorzec	**Tekst**	**W jaki sposób wzorzec występuje w tekście?**
para	*opera*	z błędem
para	*aparat*	dokładnie
kran	*karawana*	
sport	*bezspornie*	
ryt	*zakryty*	
sofa	*solanka*	

27.2.

Może się zdarzyć, że wzorzec występuje w tekście więcej niż raz: na przykład w słowie *rabarbar* wzorzec *bar* występuje dwukrotnie. Wystąpienia mogą się częściowo nakładać: wzorzec *issi* występuje dwukrotnie w *mississippi*.

Podaj przykład tekstu o długości 8 oraz wzorca o długości 4, dla których wzorzec występuje w tekście (dokładnie) przynajmniej trzykrotnie.

27.3.

Podaj algorytm (w pseudokodzie lub wybranym języku programowania), który dla danego wzorca i danego tekstu, rozstrzygnie, czy wzorzec występuje w tekście (dokładnie lub z błędem).

Algorytm powinien wypisywać TAK, jeśli wzorzec występuje, NIE — w przeciwnym wypadku.

Dane:

dodatnie liczby całkowite m i n, $n \geq m$

wzorzec[1..m], tekst[1..n], napisy złożone z małych liter alfabetu łacińskiego

Wynik:

słowo „TAK”, jeśli wzorzec występuje w tekście (dokładnie lub z błędem), zaś słowo „NIE”, jeśli nie występuje.

Zadanie 28.

Wiązka zadań *Słabe palindromy*

Niech $W[i,j]$ oznacza część słowa W, zaczynającą się na i-tej literze, a kończącej na j-tej literze W. Na przykład dla słowa $W = ABCDEF$ mamy $W[3,5] = CDE$. Z kolei $W[i]$ oznaczać będzie i-tą literę słowa W, czyli $W[i]=W[i,i]$.

Uwaga: W treści zadania nie będziemy odróżniać pojedynczych znaków od całych napisów, cojest charakterystyczne dla niektórych języków programowania.

Niech W będzie złożonym z liter A i B słowem o długości m. Słowo W nazywamy *słabym A-palindromem*, jeśli spełnia ono jeden z dwóch warunków:

1. W = „A”, czyli W jest słowem złożonym jedynie z litery A;
2. długość słowa m jest liczbą parzystą oraz spełnione są jednocześnie poniższe warunki:
   a. $W[1] = W[m]$
   b. $W[1,m/2]$ lub $W[m/2+1,m]$ jest słabym A-palindromem.

Przykłady

1. Jedynym dwuliterowym słabym A-palindromem jest AA.
2. Słowa $W_1=AAAA$ oraz $W_2=AABA$ są słabymi A-palindromami, oba spełniają warunek 2a (gdyż $W_1[1]=W_1[4]=A$ oraz $W_2[1]=W_2[4]=A$). Ponadto $W_1[1,2]=W_2[1,2]=AA$ jest także słabym A-palindromem, co zapewnia spełnienie warunku 2b zarówno dla W_1 jak i W_2.
3. Słowo $W=AAAAAA$ nie jest słabym A-palindromem, ponieważ nie jest spełniony warunek 2b. Słowo W ma długość m=6, co oznacza, że słowa $W[1,m/2]$ oraz $W[m/2+1,m]$ mają długość nieparzystą równą 3. W rezultacie żadne ze słów: $W[1,m/2]= AAA$, $W[m/2+1,m]= AAA$ nie jest słabym A-palindromem.
4. Słowo $AAABBAAA$ nie jest słabym A-palindromem, natomiast $AABABBBA$ jest słabym A-palindromem.

28.1.

Uzupełnij poniższą tabelę, ustalając, które słowa spełniają warunki 2a i 2b. oraz które są słabymi A-palindromami. W przypadku słabych A-palindromów w kolumnie *Uzasadnienie* wskaż tę połowę słowa, która zapewnia spełnienie warunku 2b (jeśli obie połowy zapewniają spełnienie 2b, to wystarczy wskazać tylko jedną). Dla słów, które nie są słabymi A-palindromami, kolumnę *Uzasadnienie* zostaw pustą.

Słowo	*Czy jest spełniony warunek*		*Czy słowo jest słabym A-palindromem?*	*Uzasadnienie*
	2a?	*2b?*		
AABAABAA	tak	tak	tak	AABA
AAABBAAA	tak	nie	nie	
AAABBAAB	nie	nie	nie	
AAAABBBB				
ABBBABAA				
ABAAAABA				
AAAAAAAAAA				

28.2.

Rozważmy mające różną długość słabe A-palindromy, w których występuje najmniej liter *A*. Wiemy, że jest tylko jeden słaby A-palindrom jednoliterowy równy *A* oraz jeden słaby A-palindrom dwuliterowy równy *AA*. Nietrudno sprawdzić, że wszystkie czteroliterowe słabe A-palindromy to: *AAAA, ABAA, AABA*. Zatem najmniejsza liczba liter A w słabym A-palindromie o długości 1 jest równa 1, najmniejsza liczba liter A w słabym A-palindromie o długości 2 jest równa 2, a najmniejsza liczba liter A w słabym A-palindromie o długości 4 jest równa 3.

Uzupełnij puste pola w poniższym zestawieniu:

m	*najmniejsza liczba liter A słabego A-palindromu o długości m*	*słaby A-palindrom o długości m i najmniejszej liczbie liter A*
2	2	AA
4	3	
8		

Uzupełnij puste pola w poniższej tabeli, podając najmniejszą liczbę liter A w słabych A-palindromach o podanych poniżej długościach, uzupełniając puste pola w poniższej tabeli.

m	*najmniejsza liczba liter A słabego A-palindromu o długości m*
16	
32	
2^{10}	
2^{20}	

28.3.

Podaj algorytm sprawdzający, czy podane na wejściu słowo *W* jest słabym A-palindromem. Twój algorytm powinien działać zgodnie z następującą specyfikacją:

Specyfikacja

Dane:

słowo *W*

Wynik:

„Tak”, gdy *w* jest słabym A-palindromem,

„Nie”, gdy w nie jest słabym A-palindromem.

Zadanie 29.

Wiązka zadań *Alfabet kulkowy*

W Kulkolandii do zapisu wiadomości używa się kulek o dwóch kolorach, czarnym i białym. Każda litera alfabetu jest zapisywana za pomocą pewnej liczby kulek odpowiednio uporządkowanych tak, jak to przedstawiono w poniższej tabeli. Zauważ, że zapis każdej litery kończy się dwiema czarnymi kulkami:

A	●●	I	○○●○●●	R	●○○○○●●
B	○●●	J	○●○○●●	S	●○○●○●●
C	○○●●	K	●○○○●●	T	●○●○○●●
D	●○●●	L	●○●○●●	U	○○○○○○●●
E	○○○●●	M	○○○○○●●	W	○○○○●○●●
F	○●○●●	N	○○○●○●●	X	○○○●○○●●
G	●○○●●	O	○○●○○●●	Y	○○●○○○●●
H	○○○○●●	P	○●○○○●●	Z	○●○○○○●●

29.1.

Odczytaj poniższą wiadomość zapisaną alfabetem kulkowym:

○○○○○●●●●●○●○○●●○○○○○○●●●○○○○●●●●

29.2.

Do dyspozycji masz trzy operacje:

- *pusty(ciąg)*, która inicjuje pusty ciąg kulek;
- *pobierz(ciąg),* która usuwa pierwszą kulkę z niepustego ciągu kulek i podaje ją jako wynik;
- *dołącz(ciąg, kulka)*, która do podanego ciągu dołącza na końcu wskazaną kulkę.

Zapisz algorytm (w pseudokodzie lub w wybranym języku programowania), który z danego ciągu kulek *wiadomosc* usuwa początkowy fragment odpowiadający pierwszej literze i zapisuje go jako ciąg kulek w zmiennej *litera*.

Specyfikacja
Dane:
wiadomosc — ciąg kulek z zakodowanym napisem
Wynik:
litera — ciąg kulek odpowiadający pierwszej literze z zakodowanego napisu
wiadomosc — ciąg kulek pozostałych po pobraniu kulek odpowiadających pierwszej literze
Uwaga: połączenie wynikowych *litera* i *wiadomosc* daje w rezultacie wejściową *wiadomosc*

29.3.

Do dyspozycji masz operacje:

- *pusty(tekst)*, która inicjuje pusty tekst;
- *pobierz_litere(ciąg)*, która usuwa z niepustego ciągu kulkowego z zapisaną wiadomością kulki odpowiadające pierwszej literze, a jako wynik podaje literę

odpowiadającą usuniętym kulkom;
- *dopisz(tekst, litera)*, która do podanego tekstu (może on być pusty) dołącza na końcu wskazaną literę;
- *czy_sa_kulki(ciąg)*, podającą wartość PRAWDA, gdy w podanym ciągu znajduje się przynajmniej jedna kulka, a wartość FAŁSZ, gdy w podanym ciągu nie ma żadnej kulki.

Zapisz algorytm (w pseudokodzie lub w wybranym języku programowania), który dekoduje napis zapisany kulkami.

Specyfikacja
Dane:
ciąg — ciąg kulek z zakodowaną wiadomością
Wynik:
wiadomosc — tekst z dekodowaną wiadomością

Zadanie 30.
Wiązka zadań *Szyfrowanie*

Szyfrowanie z kluczem k (k — liczba całkowita większa lub równa 0) polega na zastąpieniu każdego znaku wiadomości jawnej innym znakiem o pozycji przesuniętej w alfabecie o k znaków względem znaku szyfrowanego. Przy szyfrowaniu znaku należy postępować w sposób cykliczny, tzn. po osiągnięciu końcowego znaku alfabetu należy powrócić na jego początek. Rozważamy tylko wielkie litery alfabetu angielskiego, tj. litery o kodach dziesiętnych ASCII od 65 (dla znaku A) do 90 (dla znaku Z).

30.1.

Mając do dyspozycji funkcję *kod*(x), która dla danego znaku x podaje dziesiętny kod ASCII tego znaku (np. *kod*(A) = 65), oraz funkcję *znak*(y), która dla danego dziesiętnego kodu ASCII podaje znak odpowiadający temu kodowi (np. *znak*(77) = M), napisz funkcję *szyfruj*(zn,k), szyfrującą znak *zn* szyfrem z kluczem k.

Przykład działania funkcji: *szyfruj*(M,327) = B.

30.2.

Aby utrudnić proces deszyfrowania, postanowiono dla każdego znaku w słowie zastosować inny klucz. Kluczem znaku w słowie będzie numer jego pozycji w tym słowie.

Przykład: w słowie BIT, dla znaku B mamy k=1, dla I — k=2, a dla T — k=3. Zaszyfrowane słowo BIT to CKW.

Uzupełnij poniższą tabelę. Zaszyfruj opisanym sposobem podane słowo jawne oraz odszyfruj słowo zaszyfrowane.

Słowo jawne	**Słowo zaszyfrowane**
INFORMATYKA	
	LQPTZZLZ

30.3.

Dane jest słowo S zaszyfrowane sposobem opisanym w zadaniu 2. Podaj algorytm (w postaci pseudokodu lub kodu wybranego języka programowania) deszyfrujący słowo S.

Dane:

- n — liczba znaków w słowie S (n>0)
- $S[1..n]$ — zaszyfrowane słowo

Wynik:

- $J[1..n]$ — słowo jawne, które po zaszyfrowaniu daje słowo S

Zadanie 31.

Wiązka zadań *Szyfr skokowy*

Szyfrem skokowym z kluczem k>0 nazywać będziemy kodowanie tekstu opisane przez poniższy algorytm, w którym danymi na wejściu są: k — liczba całkowita dodatnia oraz W — tekst.

Algorytm:

```
n ← dlugosc(W)
m ← n div k
jeżeli n mod k≠0
        m←m+1
dla i=1,2,…,m wykonuj
        j←i
        dopóki j ≤ n wykonuj
                wypisz W[j]
                j ← j+m
```

Uwaga: Przyjmujemy, że:

- dostępna jest funkcja *dlugosc*, która zwraca długość słowa będącego jej argumentem;
- kolejne znaki słowa W o długości n oznaczamy przez $W[1]$, $W[2]$,…, $W[n]$.

Przykład

Dla k=3 i tekstu SZYFROWANIE algorytm wypisze tekst SRNZOIYWEFA.

31.1.

Dla poniższych wartości k i W podaj tekst wypisany przez algorytm:

- k=3, W=ZADANIE1JESTŁATWE

  Tekst wypisany przez algorytm:.....
- k=4, W=ZADANIE1JESTPROSTE

  Tekst wypisany przez algorytm:.....

31.2.

Podaj teksty W, dla których powyższy algorytm wypisze na wyjściu podane wartości:

- k=3, W= .....

Tekst wypisany przez algorytm: UDOMEWIKAEOĆMD

- k=4, W= .....

  Tekst wypisany przez algorytm: DRJTOZEBES

31.3.

Podaj algorytm deszyfrowania tekstu zakodowanego szyfrem skokowym zgodny z poniższą specyfikacją:

Specyfikacja

Dane:

k — liczba całkowita dodatnia

X — tekst

Wynik:

W — tekst, którego szyfr skokowy z kluczem k jest równy W.

Przykład

Dla k=3 i X=SRNZOIYWEFA algorytm powinien zwrócić na wyjściu tekst W=SZYFROWANIE.

31.4.

Szyfrem mieszanym z kluczem k>0 nazywać będziemy następujący sposób kodowania tekstu W o długości n:

1. Dzielimy tekst W na k części tak, że wszystkie części poza ostatnią mają tę samą długość m równą zaokrągleniu w górę liczby n / k. Ostatnia część ma długość mniejszą lub równą m.
2. Tak uzyskane części wpisujemy w kolejnych wierszach prostokątnej tabeli o k wierszach i m liczbie kolumn.
3. Wypisujemy kolejne kolumny, zaczynając od pierwszej, zgodnie z następującą zasadą: kolumny o nieparzystym numerze wypisujemy z góry na dół, a kolumny o parzystym numerze z dołu do góry.

Przykład

Dla k=3 i tekstu W=SZYFROWANIE uzyskamy tabelę:

S	Z	Y	F
R	O	W	A
N	I	E	

a zaszyfrowana tekst ma postać SRNIOZYWEAF.

Podaj algorytm szyfrowania tekstu zakodowanego szyfrem mieszanym zgodny z poniższą specyfikacją:

Specyfikacja

Dane:

k — liczba całkowita dodatnia

W — tekst

Wynik:

X — szyfr mieszany tekstu W z kluczem k.

Zadanie 32.
Wiązka zadań *Kompresja*

Dany jest napis złożony z małych liter alfabetu angielskiego, który kompresujemy w następujący sposób: jeżeli w napisie występują kolejno po sobie dwa identyczne ciągi kolejnych znaków, na przykład *piripiri*, to możemy zastąpić je jednokrotnym wystąpieniem tego ciągu, ujętym w nawiasy okrągłe, np. *(piri)*.

Na przykład *xyzxyz* można zapisać jako *(xyz)*, zaś *rabarbar* jako *ra(bar)*.

32.1.

Niektóre fragmenty napisów znajdujących się w tabelce zostały skompresowane podaną powyżej metodą. Uzupełnij tabelkę, odtwarzając oryginalne napisy.

Napis skompresowany	**Napis oryginalny**
a(cd)a	acdcda
(pur)owy	
(z)(zz)	
(ab)a(abcd)	

32.2.

Podaj algorytm (w pseudokodzie lub wybranym języku programowania), który mając zapisany w tablicy pewien napis, sprawdzi czy jest on (w całości) dwukrotnym powtórzeniem pewnego fragmentu, a jeśli tak, wypisze skompresowany napis w postaci *(fragment)*. Jeśli napis wejściowy nie jest powtórzeniem, na wyjście nie należy nic wypisywać.

Dane:

- liczba całkowita $n>0$ oraz tablica *napis[1..n]*, zawierająca napis złożony z małych liter alfabetu angielskiego

Wynik:

- jeżeli napis wejściowy jest dwukrotnym powtórzeniem tego samego fragmentu, wynikiem są kolejne znaki napisu skompresowanego.

32.3.

Dany jest napis, w którym niektóre fragmenty zostały skompresowane podaną powyżej metodą. Podaj algorytm (w pseudokodzie lub wybranym języku programowania), który mając na wejściu skompresowany napis zapisany w tablicy, wypisze na wyjście kolejne litery napisu oryginalnego.

Dane**:**

dodatnia liczba całkowita *n* oraz tablica *napis[1..n]*, zawierająca pewien napis skompresowany metodą opisaną wcześniej.

Wynik:

kolejne znaki oryginalnego napisu, którego skompresowana wersja jest w tablicy *napis.*

Uwaga: W napisie skompresowanym nie ma zagnieżdżeń nawiasów, czyli wewnątrz pary nawiasów nie wystąpią inne nawiasy.

Zadanie 33.

Wiązka zadań *Wędrówka po planszy*

Wędrowiec podróżuje po kwadratowej planszy rozmiaru $n \times n$. Swoją wędrówkę rozpoczyna na dowolnym polu **pierwszej kolumny** planszy, a na koniec powinien dotrzeć do **ostatniej kolumny**. Będąc w kolumnie $j<n$, wędrowiec może w jednym ruchu przenieść się do kolumny $j+1$ (nie może przenieść się do żadnej innej kolumny). Wartość każdego pola planszy jest liczbą całkowitą. Wartość pola w i-tym wierszu i j-tej kolumnie na planszy A oznaczać będziemy przez $A[i, j]$.

Rozważamy 3 typy wędrowca:

- skaczący, który z pola w kolumnie $j<n$ może przeskoczyć na dowolne pole w kolumnie $j+1$;
- spadający, który z pola $A[i, j]$ możne przenieść się tylko na pola $A[k, j+1]$ takie, że $i \le k \le n$;
- chodzący, który z pola $A[i, j]$ może przeskoczyć tylko na pola $A[k, j+1]$ takie, że $k=i$ lub $k=i-1$ lub $k=i+1$ oraz $1 \le k \le n$.

Przykład

Rozważmy planszę rozmiaru 10×10. Jeżeli bieżącą pozycją wędrowca jest pole $A[3,4]$, to

- **wędrowiec skaczący** może w jednym ruchu przenieść się do pól $A[1,5]$, $A[2,5],\ldots,A[10,5]$,
- **wędrowiec spadający** może przenieść się do pól $A[3,5]$, $A[4,5],\ldots,A[10,5]$,
- **wędrowiec chodzący** może przenieść się do pól $A[2,5]$, $A[3,5]$, $A[4,5]$.

Rozważmy następujący algorytm, opisujący trasę jednego z typów wędrowców, dla poniższych danych:

Dane: n — liczba naturalna większa od 1;
A — tablica rozmiaru $n \times n$ wypełniona liczbami całkowitymi.

Uwaga: W poniższym algorytmie $B[0...n+1,1..n]$ jest tablicą, której wiersze mają numery $0,1,\ldots,n,n+1$, a kolumny $1,2,\ldots,n-1, n$.

Algorytm:

```
dla i=1,2,…,n wykonuj
        jeżeli A[i,1]>0
                B[i,1] ← 1
        w przeciwnym razie
                B[i,1] ← 0
dla j=2,3,…,n wykonuj
        B[0,j] ← 0
        B[n+1,j] ← 0
        dla i=1,2,…,n wykonuj
                jeżeli A[i,j] ≤0
                        B[i,j] ← 0
                w przeciwnym razie
                        jeżeli B[i – 1, j – 1]=1 lub B[i, j – 1]=1 lub B[i + 1, j – 1]=1
                                B[i,j] ← 1
                        w przeciwnym razie
                                B[i,j] ← 0
d ← 0
dla i=1,2,…,n wykonuj
        jeżeli B[i,n]=1
                d ← 1
zwróć d
```

33.1.

Rozważmy działanie algorytmu dla n=5 oraz następującej zawartości tablicy A:

	1	2	3	4	5
1	–2	–1	4	7	8
2	–3	2	3	–10	–2
3	1	–4	–1	5	–5
4	–2	–1	–2	–3	9
5	–1	–5	1	–4	1

Podaj końcową zawartość kolumn 1,2,…,5 **tablicy B** oraz wartość zwracaną przez algorytm_1 dla powyższych danych.

Tablica B:

	1	2	3	4	5
0					
1					
2					
3					
4					
5					
6					

Wartość zwracana przez algorytm: .....

33.2.

Uzupełnij specyfikację podanego powyżej algorytmu.

Specyfikacja

Dane:

n — liczba naturalna większa niż 1

A — plansza rozmiaru $n \times n$ wypełniona liczbami całkowitymi.

Wynik:

1 — jeśli istnieje trasa wędrowca typu .....,

i prowadząca tylko przez pola o wartościach ..... .

0 — w przeciwnym przypadku

33.3.

Wartością trasy wędrowca nazywamy sumę liczb zapisanych na polach planszy, które wędrowiec odwiedza w trakcie trasy. Podaj algorytm zgodny z poniższą specyfikacją.

Specyfikacja

Dane:

n — liczba naturalna większa niż 1

A — plansza rozmiaru $n \times n$ wypełniona liczbami całkowitymi.

Wynik:

Największa wartość trasy wędrowca typu skaczącego zaczynającej się w pierwszej kolumnie i kończącej się w ostatniej kolumnie.

Przykład

Dla n=5 oraz zawartości planszy podanej w zadaniu 1 algorytm powinien zwrócić wartość 23; trasa wędrowca o największej wartości prowadzi przez pola $A[3,1]$, $A[2,2]$, $A[1,3]$, $A[1,4]$ i $A[4,5]$.

33.4.

Podaj algorytm zgodny z poniższą specyfikacją.

Specyfikacja

Dane:

n — liczba naturalna większa niż 1

A — plansza rozmiaru $n \times n$ wypełniona liczbami całkowitymi.

Wynik:

1 — jeśli istnieje trasa wędrowca spadającego zaczynająca się w pierwszej kolumnie, kończąca się w ostatniej kolumnie i przechodząca wyłącznie przez pola o wartościach nieujemnych;

0 — w przeciwnym przypadku.

Przykład

Dla *n*=5 oraz zawartości planszy podanej w zadaniu 1 algorytm powinien zwrócić wartość 0, gdyż trasa spełniająca podane warunki musi zaczynać się w polu *A*[3,1], z którego można przejść tylko do pól ujemnych w drugiej kolumnie. Dla *n*=5 i poniższej zawartości planszy algorytm powinien zwrócić wartość 1; trasa prowadząca tylko przez pola dodatnie może przechodzić np. przez pola: *A*[1,1], *A*[2,2], *A*[2,3], *A*[3,4] i *A*[4,5].

	1	2	3	4	5
1	2	–1	4	7	8
2	–3	2	3	–10	–2
3	1	–4	–1	5	–5
4	–2	–1	–2	3	9
5	–1	–5	–6	–4	1

1.3. Praktyka w teorii

Zadanie 34.

Bajtek i Bituś poznali starożytny chiński sposób testowania pierwszości liczby naturalnej *n*:

liczba *n* musi spełniać równanie

$$2^n \bmod n = 2,$$

gdzie *mod* oznacza operator dzielenia modulo, czyli resztę z dzielenia całkowitego.

Uwaga: Starożytny chiński sposób bywa zawodny, istnieją liczby, które spełniają to równanie, a nie są pierwsze; natomiast jeśli równanie nie zachodzi, to *n* jest na pewno złożona.

Chłopcy postanowili przetestować chińską metodą kolejne liczby n>=2, przeprowadzając obliczenia tylko na takich liczbach, które można reprezentować bez znaku na 8 bitach. co znaczy, że na każdym etapie obliczeń stosowali tylko liczby całkowite z zakresu od 0 do 255.

Bajtek obliczał potęgi 2^n dla kolejnych wartości *n* i dla obliczonej wartości wyznaczał resztę z dzielenia przez *n*.

Bituś skorzystał z praw arytmetyki modularnej. W każdym z kolejnych kroków obliczania potęgi, wynik iloczynu brał modulo *n*. Na przykład dla *n*=3 obliczenia wykonywał następująco:

$$2^3 \bmod 3 = ((2*2) \bmod 3)*2 \bmod 3 = 1*2 \bmod 3 = 2.$$

Który z nich mógł przetestować pierwszość liczb w większym zakresie?

Podaj możliwie największą wartość *n*, dla której możliwe było przeprowadzenie testu pierwszości metodą każdego z chłopców.

W obliczeniach Bajtka n_{max} = ……

W obliczeniach Bitusia n_{max} = ……

Zadanie 35.

Jeden ze sposobów zapisania niezerowej liczby rzeczywistej *x* w pamięci komputera polega na:

1) przedstawieniu tej liczby w postaci iloczynu trzech liczb:

   $(-1)^S$ — przy czym *S* jest równe 0 lub 1 i nazywa się *znakiem*,

   M — *mantysy*, która jest liczbą z przedziału [1;2),

   2^C — przy czym *C* jest liczbą całkowitą zwaną *cechą*,

$$x = (-1)^S * M * 2^C.$$

2) zapisaniu w pamięci oddzielnie: znaku, mantysy i cechy.

Przykład

$$5.5 = 2.75*2^1 = \underbrace{1.375}_{\text{mantysa}}*2^{\overset{\text{cecha}}{\textcircled{2}}}$$

Przyjmijmy, że każdą liczbę rzeczywistą zapisujemy na **8 bitach**.

Najstarszy bit (pierwszy z lewej) jest **bitem znaku**: 0 oznacza liczbę dodatnią, 1 — ujemną.

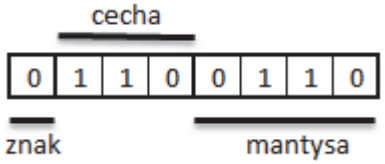
Następne 3 bity reprezentują **cechę** — liczbę całkowitą z zakresu od –4 do 3, reprezentowaną przez trzy cyfry binarne następująco:

-4	000
-3	001
-2	010
-1	011

0	100
1	101
2	110
3	111

Do zapisu **mantysy** pozostają ostatnie 4 bity. Część całkowita mantysy będzie zawsze równa 1, więc wystarczy zapisać tylko jej część ułamkową. Ułamek przedstawiamy w postaci binarnej i zapisujemy cztery pierwsze cyfry rozwinięcia, o wagach: 2^{-1}, 2^{-2}, 2^{-3} i 2^{-4}.

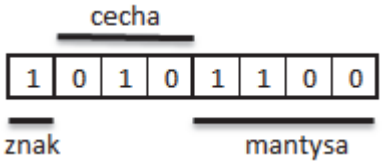
Przykład



Bit znaku ma wartość 0, więc liczba jest dodatnia
$C = 110_2 - 4 = 6 - 4 = 2$
$M = \mathbf{1}.0110_2 = 1 + 0.25 + 0.125 = 1.375$
$x = M*2^C = 1.375*2^2 = \mathbf{5.5}$

35a.

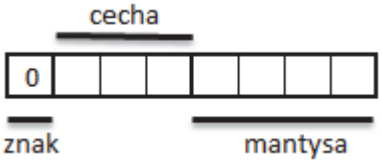
Oceń, czy podane poniżej zdania dotyczące liczby rzeczywistej zapisanej obok w postaci binarnej są prawdziwe, czy fałszywe, stawiając znak X w odpowiedniej kolumnie poniższej tabeli:



		P	F
A	Liczba jest dodatnia.		
B	Cecha ma wartość dziesiętną równą -2.		
C	Mantysa ma wartość dziesiętną równą 0.75.		
D	Liczba ma wartość dziesiętną równą -0.4375.		

35b.

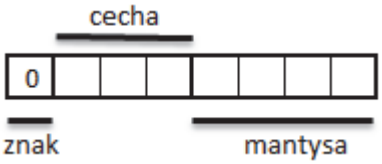
Jaką największą liczbę rzeczywistą x można zapisać, wykorzystując 8 bajtów w sposób opisany wyżej? Wypełnij bity reprezentacji binarnej tej liczby i podaj jej wartość dziesiętną.



C = ...
M = ...
x = ...

35c.

Jaką najmniejszą liczbę nieujemną można zapisać w opisanej wyżej reprezentacji 8-bitowej? Wypełnij bity jej reprezentacji binarnej i podaj jej wartość dziesiętną.



C = …….
M = …….
x = ……...

Zadanie 36.

Dane są tabele `Uczniowie` i `Oceny`

Uczniowie:

Id_u	Imie	Nazwisko
1	Marek	Wolski
2	Hanna	Wieczorek
3	Irena	Iwanicka
4	Wojciech	Moscicki
5	Janina	Idzik
6	Arkadiusz	Czarnecki

Oceny

Id_u	Wynik
1	2
1	3
2	5
2	4
3	5
4	2
4	5
5	2
6	4
6	3

```
SELECT Uczniowie.Nazwisko
FROM Uczniowie
JOIN Oceny ON Uczniowie.Id_u = Oceny.Id_u
GROUP BY Uczniowie.Nazwisko
HAVING Avg(Oceny.Wynik)>=4.5;
```

Zaznacz nazwiska, które pojawią się w wyniku powyższego zapytania.

	T	N
Wolski		
Wieczorek		
Iwanicka		
Moscicki		
Idzik		
Czarnecki		

Zadanie 37.

Dana jest tabela `Produkty`:

Id	**Nazwa**	**Cena**
1	banany	3,20
2	jabłka	2,60
3	winogrona	7,20
4	gruszki	4,10
5	orzechy	5,30
6	pomarańcze	4,80
7	maliny	6,60
8	śliwki	4,10

```
SELECT Nazwa
FROM Produkty
ORDER BY Cena DESC;
```

Wśród pierwszych trzech wierszy wyniku dla powyższego zapytania pojawią się:

	T	N
gruszki		
orzechy		
pomarańcze		
maliny		
śliwki		

Zadanie 38.

Dana jest tabela `Agenci`:

ID_agenta	Obszar_dzialania
A001	Katowice
A002	Katowice
A003	Warszawa
A004	Warszawa
A005	Kraków
A006	Kraków
A007	Katowice

```
SELECT obszar_dzialania, COUNT(*)
FROM Agenci
GROUP BY obszar_dzialania;
```

Wynik powyższego zapytania to:

	T	N
Katowice 1 Warszawa 2 Kraków 3		
Katowice 3 Kraków 2 Warszawa 2		
Katowice A001 Kraków A005 Warszawa A003		
A00* 3		

Zadanie 39.

Wiązka zadań *Praktyka w teorii — Grafika*

Na stronie internetowej aquaparku postanowiono umieścić zdjęcia zachęcające do odwiedzin tego miejsca. Zarządcy aquaparku zależy, aby ze strony internetowej mogły korzystać także osoby mające jedynie dostęp do łącza internetowego niskiej prędkości (np. 1 Mbit/s). Jedno ze zdjęć o nazwie `aquapark1` zostało przekształcone za pomocą programu graficznego i zapisane jako `aquapark2`. W tabeli przedstawiono oba zdjęcia wraz z ich parametrami.



aquapark1	
typ obrazu:	jpeg
data:	14.08.2014
rozmiar:	3072 x 2304
głębia w bitach:	24bits
rozmiar pliku:	1,17 MB



aquapark2	
typ obrazu:	jpeg
data:	14.08.2014
rozmiar:	1024 x 768
głębia w bitach:	8bits
rozmiar pliku:	195 KB

39.1.

Który obraz powinien zostać dodany do galerii zdjęć na stronie aquaparku? Uzasadnij swój wybór.

39.2.

Dział promocji aquaparku zamierza przygotować do druku folder, w którym umieści różne zdjęcia. Wybierz z poniższej listy format pliku graficznego, który zapewni najlepszą jakość druku. Uzasadnij swój wybór.

TIFF	JPEG	GIF

39.3.

Ile pamięci zajmie bitmapa 1024 x 768 pikseli, jeśli zapisano ją w systemie RGB, przeznaczając na każdą składową 8 bitów? Wynik podaj w kilobajtach.

Zadanie 40.

Wiązka zadań *Praktyka w teorii*

Dla następujących zdań **zaznacz znakiem X**, która odpowiedź jest prawdziwa (P), a która jest fałszywa (F).

40.

W kolumnach `A`, `B`, `C`, `D` arkusza kalkulacyjnego w wierszach od 1 do 200 są wpisane oceny uczniów w postaci liczb całkowitych z przedziału <1,6> z następujących przedmiotów: język polski, język angielski, geografia i biologia.

Przykład

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**
1	1	4	5	6					
2	3	3	1	5					
3	2	5	2	2					
4	5	2	5	3					
……									
200	6	2	5	3					

Aby w poszczególnych kolumnach `A`, `B`, `C`, `D` obliczyć liczbę ocen bardzo dobrych (o wartości liczbowej 5) i umieścić wyniki w komórkach `G1:J1`, można w komórce `G1` zastosować następującą formułę i skopiować ją do pozostałych komórek:

	P	F
`LICZ.JEŻELI(B$1:B$200;"=5").`		
`LICZ.JEŻELI(A$1:A$200;"=5").`		
`LICZ.JEŻELI($A$1:$A$200;"=5").`		
`LICZ.JEŻELI(A1:A200;"=5").`		

41.

Numer PESEL ma 11 cyfr. Dla osób urodzonych w latach 2000-2009 pierwszą cyfrą numeru PESEL jest zero. Aby dane w pliku tekstowym, zawierające kolumnę z numerem PESEL, zostały poprawnie zaimportowane do arkusza kalkulacyjnego (jako ciągi jedenastu znaków), należy podczas importu nadać tej kolumnie format:

	P	F
tekstowy.		
liczbowy.		
ogólny, a po zaimportowaniu zmienić go na format tekstowy.		
ogólny, a po zaimportowaniu zmienić go na specjalny format o nazwie „Numer PESEL”.		

Zadanie 42.

Wiązka zadań *Podział tablicy*

Rozważamy następujący algorytm.

Dane:

tablica liczb naturalnych $T[1..n]$

Algorytm:

```
        x ← T[1]
        i ← 0
        j ← n+1
        wykonuj
            wykonuj
                j ← j-1
(*)         dopóki T[j] > x
            wykonuj
                i ← i+1
(**)        dopóki T[i] < x
            jeżeli  i < j
(***)           zamień(T[i], T[j])
            w przeciwnym razie
                zakończ
```

Uwaga: funkcja *zamień*(T[i], T[j]) zamienia miejscami wartości T[i] oraz T[j].

42.1.

Przeanalizuj działanie algorytmu i podaj łączną liczbę operacji porównania, jakie zostaną wykonane w wierszach oznaczonych (*) i (**) dla danych zapisanych w poniższej tabeli:

Tablica T	**Liczba operacji porównania wykonanych w wierszu oznaczonym (*)**	**Liczba operacji porównania wykonanych w wierszu oznaczonym (**)**
4, 2, 5, 8, 1, 9, 7, 6, 3		
5, 4, 3, 2, 1, 6, 7, 8, 9, 10		
1, 2, 3, ... , 100		
100, 99, 98, ... , 1		

42.2.

Przeanalizuj działanie algorytmu i podaj łączną liczbę operacji zamiany, jakie zostaną wykonane w wierszu oznaczonym (***) dla danych zapisanych w poniższej tabeli:

Tablica T	**Liczba operacji zamiany wykonanych w wierszu oznaczonym (***)**
4, 2, 5, 8, 1, 9, 7, 6, 3	
5, 4, 3, 2, 1, 6, 7, 8, 9, 10	
1, 2, 3, ... , 100	
100, 99, 98, ... , 1	

Zadanie 43.

Wiązka zadań *Smartfon*

Rozważmy następujące aplikacje:

- serwis pogodowy,
- katalog książek biblioteki szkolnej,
- elektroniczny dziennik lekcyjny,
- edytor tekstu,
- serwis informacji turystycznej Wrocławia,
- wyszukiwanie optymalnej trasy samochodowej,
- arkusz kalkulacyjny,
- kompilator języka programowania,
- system komputerowego składu tekstu.

43.1.

Wskaż wśród powyższych aplikacji dwie, dla których przydatna jest informacja o położeniu geograficznym użytkowników korzystających z urządzeń mobilnych (np. smartfonów). Dla każdej z wybranych aplikacji opisz w jednym lub dwóch zdaniach sposób wykorzystania danych o lokalizacji użytkowników.

43.2.

Wskaż wśród powyższych aplikacji co najmniej dwie, których przydatność jest istotnie ograniczona w sytuacji, gdy użytkownik ma jedynie dostęp do urządzenia mobilnego (np. smartfon, tablet). Dla każdej z wybranych aplikacji uzasadnij swój wybór w co najwyżej dwóch zdaniach.

43.3.

Zdjęcia wykonywane na smartfonie Franka gromadzone są na jego koncie w chmurze obliczeniowej, o ile smartfon jest podłączony do Internetu. W przypadku braku dostępu do Internetu zdjęcia gromadzone są w pamięci podręcznej. Przy obecnych ustawieniach smartfonu często występuje problem braku miejsca w pamięci podręcznej, uniemożliwiający robienie dużej liczby zdjęć. Franek chciałby zmienić ustawienia tak, aby możliwe było zachowanie w pamięci większej liczby zdjęć.

Wśród podanych niżej sposobów wskaż te, które mogą pomóc w rozwiązaniu problemu Franka (zaznacz znakiem X która odpowiedź jest prawdziwa, a która jest fałszywa):

	P	F
zmiana formatu zapisu zdjęć na mapę bitową		
zmniejszenie rozdzielczości zapisywanych zdjęć		
zastosowanie kodów korekcji CRC, opartych na bitach parzystości		
obniżenie jakości kompresji zdjęć (w formacie JPEG)		

43.4.

Franek założył konto w banku i zamierza korzystać z bankowości internetowej. Wśród poniższych funkcjonalności wskaż te, które mogą służyć zabezpieczaniu usług bankowości internetowej przed nieuprawnionym dostępem (zaznacz znakiem X która odpowiedź jest prawdziwa, a która jest fałszywa):

	P	F
protokół transferu plików FTP (File Transfer Protocol).		
protokół SSL (Secure Socket Layer).		
uwierzytelnianie użytkownika przy pomocy hasła lub PIN.		
kompresja dysku twardego.		
hasła jednorazowe generowane przez układy kryptograficzne i dostarczane kanałami informacyjnymi alternatywnymi dla Internetu.		

1.4. Test z ogólnej wiedzy informatycznej

Zadanie 44.

Zdecyduj, które z dokończeń podanego niżej zdania czynią z niego zdanie prawdziwe (P), a które fałszywe (F). Zaznacz to **znakiem X** w odpowiednich miejscach tabeli.

Liczba 100110010_2

	P	F
jest dwa razy większa od liczby $10011001_{2.}$		
jest dwa razy mniejsza od liczby $1001100100_{2.}$		
jest większa niż $512_{10.}$		
jest mniejsza niż $472_{8.}$		

Zadanie 45.

Wskaż elementy, które są niezbędne do uruchomienia komputera i załadowania systemu operacyjnego.

	P	F
procesor		
twardy dysk		
pamięć operacyjna		
monitor		

Zadanie 46.

Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe, a które jest fałszywe.

	P	F
System operacyjny przydziela zadaniom czas pracy procesora.		
System operacyjny używa zawsze tego samego systemu plików dla wszystkich urządzeń.		
W skład systemu operacyjnego wchodzi zawsze graficzny interfejs użytkownika.		
System operacyjny przydziela uruchamianym aplikacjom pamięć operacyjną.		

Zadanie 47.

Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe (P), a które jest fałszywe (F).

System plików NTFS

	P	F
nie jest obsługiwany przez system Linux.		
przechowuje informację o rozmiarze, dacie utworzenia i modyfikacji pliku oraz o ścieżce dostępu do pliku.		
uniemożliwia zapisanie pojedynczego pliku o rozmiarze powyżej 4 GB.		
umożliwia administratorowi nadawanie pojedynczym użytkownikom lub grupom użytkowników praw dostępu do plików i katalogów.		

Zadanie 48.

Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe, a które jest fałszywe.

W pewnej firmie znajdują się m.in. komputery o następujących adresach IP:

- komputer A: 10.20.30.40 / maska 255.255.0.0;
- komputer B: 10.0.0.10 / maska 255.255.255.0;
- komputer C: 1.2.3.4 / maska 255.255.255.0;
- komputer D: 1.2.3.250 / maska 255.255.255.0.

	P	F
Komputer A może być widoczny w sieci Internet pod innym adresem IP.		
Tylko dwa z wymienionych komputerów mogą mieć dostęp do sieci Internet.		
Komputery A i B znajdują się w jednej podsieci.		
Komputery C i D muszą znajdować się w jednym budynku.		

Zadanie 49.

Chmura obliczeniowa jest usługą polegającą na zdalnym udostępnieniu mocy obliczeniowej urządzeń IT, oferowaną przez zewnętrznego dostawcę. Oceń prawdziwość poniższych zdań, umieszczając znak X w odpowiedniej kolumnie tabeli.

		P	F
A	Z aplikacji i danych umieszczonych w chmurze można korzystać z dowolnej lokalizacji i dowolnego sprzętu IT umożliwiającego połączenie internetowe.		
B	Użytkownik nie jest zobowiązany do zakupu licencji na oprogramowanie używane w chmurze i udostępniane przez dostawcę, płaci jedynie za jego użycie (każdorazowo lub w formie abonamentu).		
C	Użytkownik może zdalnie instalować w przydzielonych zasobach chmury dowolne aplikacje i korzystać z nich tak jak na lokalnym komputerze.		
D	Pula zasobów użytkownika (w tym: procesory, pamięć RAM, przestrzeń dyskowa) jest elastycznie skalowana w zależności od jego potrzeb i ograniczona tylko możliwościami dostawcy.		

Zadanie 50.

HTTP Cookie jest niewielką porcją informacji wysyłaną przez witrynę internetową do przeglądarki klienta i zapisywaną w jej ustawieniach. Oceń prawdziwość poniższych zdań, umieszczając znak X w odpowiedniej kolumnie tabeli.

		P	F
A	*Cookie* zawiera polecenia, które konfigurują ustawienia przeglądarki klienta.		
B	*Cookie* umożliwia serwisowi sprawdzenie, czy klient już go odwiedzał w przeszłości, oraz zapamiętanie upodobań klienta.		
C	*Cookie* zapisane przez serwis z domeny *cwaniak.org* może być odczytane przez serwis z domeny *spryciarz.org*.		
D	Zablokowanie obsługi *cookie* w przeglądarce może spowodować utrudnienia dla użytkownika dokonującego zakupów w sklepie internetowym.		

Zadanie 51.

Czterech użytkowników założyło konta w pewnym serwisie internetowym. Wybrali następujące dane logowania:

Użytkownik A	nazwa użytkownika: *jankowalski* hasło: *1234*
Użytkownik B	nazwa użytkownika: *merkuriusz_312* hasło: *merkuriusz_312*
Użytkownik C	nazwa użytkownika: *abc* hasło: *S4o9s2n5a7*
Użytkownik D	nazwa użytkownika: *master* hasło: *password*

Przestępca usiłuje przejąć konta użytkowników, stosując w tym celu następujące dwie techniki:

- atak *brute force* (wszystkie możliwe kombinacje znaków),
- metoda psychologiczna i słownikowa (zgadnięcie hasła, sprawdzenie popularnych haseł i słów).

Wskaż, które hasła są podatne na złamanie tymi metodami, stawiając w odpowiednim polu tabeli znak X, jeśli taka podatność występuje:

	atak *brute force*	metoda słownikowa/psychologiczna
Użytkownik A		
Użytkownik B		
Użytkownik C		
Użytkownik D		

Zadanie 52.

Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe (P), a które jest fałszywe (F).

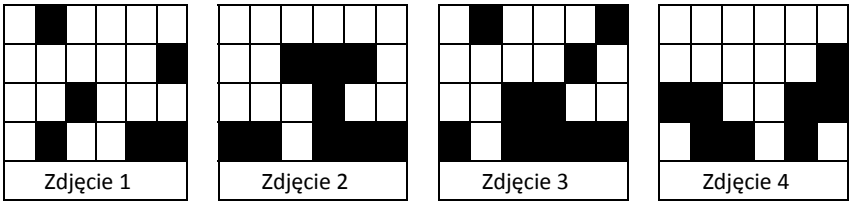
W grafice rastrowej 24-bitowe kodowanie koloru oznacza, że

	P	F
liczba kolorów w palecie barw wynosi ponad 16 milionów.		
informacje o kolorze jednego piksela zajmują 3 bajty.		
liczba kolorów jest niewystarczająca do zapisu zdjęć.		
obraz o rozmiarach 300x300 pikseli zapisany bez kompresji ma wielkość około 2,16 MB.		

Zadanie 53.

Wiązka zadań *Zegar binarny*

Pan Kowalski postanowił wyjechać na wycieczkę na wyspę Binarną. Mieszkańcy tej wyspy do zapisywania cyfr dziesiętnych używają metody obrazkowej, opartej na systemie binarnym. Podczas pobytu na wyspie Pan Kowalski wybrał się na jednodniową wycieczkę objazdową. W jej trakcie zrobił 4 poniższe zdjęcia zegara używanego na wyspie. Stan zegara podawany jest w formacie GG:MM:SS, gdzie GG to godzina, MM to minuty, a SS to sekundy (zapisane zawsze przy pomocy dwóch cyfr). Każda cyfra zapisana jest w osobnej kolumnie.



Pierwsze zdjęcie zostało zrobione w momencie rozpoczęcia wycieczki, ostatnie zdjęcie — w chwili jej zakończenia, zdjęcia 2 i 3 zostały natomiast zrobione w trakcie wycieczki.

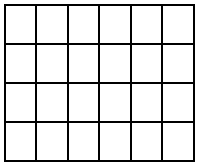
53.1.

Podaj w systemie dziesiętnym stany zegara wyświetlane na zdjęciach 1 oraz 4.

- Stan zegara na zdjęciu nr 1: ……
- Stan zegara na zdjęciu nr 4: ……

53.2.

Pan Kowalski położył się spać o godzinie 23:45:29. Zamaluj odpowiednie pola poniższej planszy tak, aby reprezentowała ona tę godzinę na zegarze używanym przez mieszkańców wyspy.



Zadanie 54.

Zaznacz znakiem X w odpowiedniej kolumnie, które zdanie jest prawdziwe (P), a które jest fałszywe (F).

Stos jest strukturą danych, która umożliwia

	P	F
bezpośredni dostęp do ostatnio zapisanego elementu.		
bezpośredni dostęp do każdego elementu stosu.		
bezpośredni dostęp do najmniejszego i największego elementu stosu.		
dodanie nowego elementu oraz usunięcie najpóźniej dodanego elementu.		

Zadanie 55.

Wiązka zadań *Zadania Zamknięte Funkcja*

Dana jest następujący algorytm F(n) dla $n \in N, n > 0$:

F(n)

jeżeli n = 1, **zwróć** 1 **i zakończ**
w przeciwnym razie zwróć F(n **div** 2) + 1

55.1.

Złożoność tego algorytmu jest

	P	F
wykładnicza.		
logarytmiczna.		
liniowa.		
kwadratowa.		

55.2.

Dla tego algorytmu zachodzi

	P	F
F(8) = 3.		
F(12) = 4.		
F(1) = 0 **lub** F(9) = 4.		
F(1) = 1 **oraz** F(9) = 3.		

Zadanie 56.

Piractwo komputerowe jest przestępstwem polegającym na:

	P	F
wykorzystywaniu oprogramowania w celu osiągnięcia korzyści majątkowej, bez licencji na jego użytkowanie.		
instalowaniu wersji demo oprogramowania, którego licencję planujemy kupić.		
rozpowszechnianiu w Internecie programów komputerowych, których licencję zakupiliśmy.		
oferowaniu za darmo w Internecie oprogramowania rozpowszechnianego na licencji freeware.		

Zadanie 57.

Zgodnie z prawem w Internecie można opublikować zdjęcie osoby:

	P	F
po uzyskaniu od niej zezwolenia.		
gdy jest to osoba powszechnie znana i zdjęcie zostało wykonane podczas pełnienia przez nią funkcji publicznych, w szczególności politycznych, społecznych, zawodowych.		
gdy osoba ta jest naszym bliskim znajomym.		
gdy stanowi ona jedynie szczegół całości takiej jak: zgromadzenie, krajobraz, publiczna impreza.		

2. Zadania praktyczne rozwiązywane z użyciem komputera

Wczytywanie danych

Dane do zadań rozwiązywanych na komputerze zapisane są w plikach tekstowych. Prosty plik tekstowy (`txt`) zawiera dane w postaci alfanumerycznej bez jakiegokolwiek formatowania. Każdy system operacyjny posiada program do otwierania plików tekstowych. W systemie Windows jest to Notatnik.

Importowanie danych do arkusza kalkulacyjnego MS Excel

W założeniu każdy wiersz pliku tekstowego ma być zaimportowany do osobnego wiersza w pliku arkusza kalkulacyjnego. Skąd jednak program ma wiedzieć, jak rozłożyć dane pomiędzy kolumnami? Spójrzmy na poniższy przykład, który przedstawia jeden wiersz z przykładowego pliku tekstowego:

```
Jan, Malinowski; 4,000; 1980-01-12
```

Dla nas jest to dość oczywiste: powyżej mamy imię i nazwisko osoby, jakąś liczbę (np. jej zarobki) i datę (np. datę urodzenia). Poszczególne pola rozdziela średnik. Powyższe dane powinny być więc zaimportowane w taki sposób:

	A	B	C	D
1	Jan	Malinowski	4,000	1980-01-12
2				

Rysunek 1- przykładowe dane po zaimportowaniu

Jednak arkusz kalkulacyjny nie potrafi analizować znaczenia danych, może je więc inaczej zaimportować. Dlatego tak ważny jest separator, który oddziela dane w wierszu.

	A	B	C	D	E
1	Jan, Malinowski; 4,000; 1980-01-12				separatorem jest tabulator
2					
3	Jan, Malinowski	4	1980-01-12		separatorem jest średnik
4					
5	Jan	Malinowski; 4	000; 1980-01-12		separatorem jest przecinek

Rysunek 2- Przykłady błędnie zaimportowanych danych

Do zaimportowania danych do arkusza służy *kreator*. Podpowiada on, jak ma być traktowany plik tekstowy, na którym pracujemy, i jak prawidłowo rozmieścić dane w komórkach. Przejdźmy więc wreszcie do arkusza kalkulacyjnego i zobaczmy, jakie pytania zadaje nam program oraz jak możemy na nie odpowiedzieć.

Rozpoczynamy import

Kreator uruchamiamy przyciskiem *Z tekstu*, znajdującym się na karcie *Dane*, w grupie *Dane zewnętrzne*.

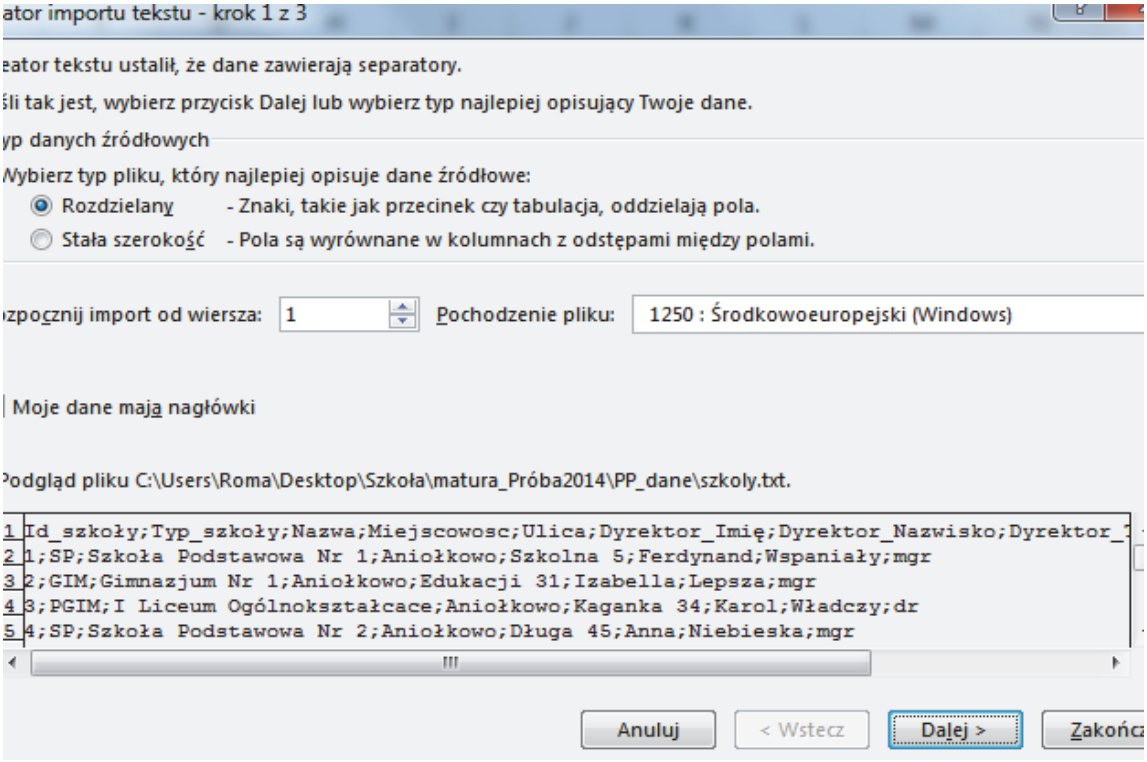


Rysunek 3 — Przycisk uruchamiający kreatora

W nowym oknie wskazujemy plik źródłowy (w naszym przypadku będzie to plik `szkoly.txt`). Załóżmy, że jest to plik zawierający informacje o różnych szkołach. Poszczególne kolumny oddzielone są znakiem **średnika**.

Przechodzimy do pierwszego kroku kreatora.

Krok 1 z 3



Rysunek 4 — Krok pierwszy importu pliku

Pierwsza decyzja dotyczy rodzaju pliku:

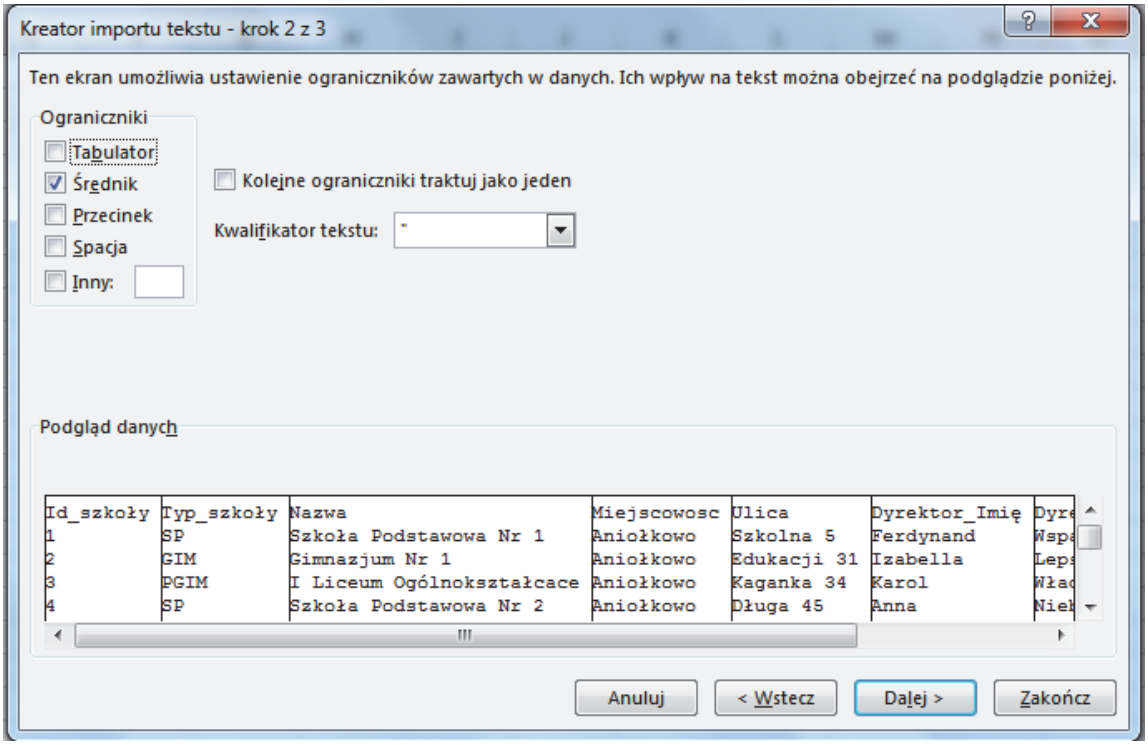
- *Czy jest to plik rozdzielany?* W pliku rozdzielanym kolejne kolumny wyróżnione są jakimś znakiem (spacja, tabulator, średnik). Arkusz, napotykając taki znak w wierszu pliku, wie, że kolejne dane ma wpisywać w następnej kolumnie.
- *Czy jest to plik z polami o stałej szerokości?* W takim pliku kolejne kolumny zajmują określoną liczbę znaków. Na przykład pierwsza kolumna zajmuje dziesięć znaków, a druga — osiem. Po wpisaniu do pierwszej kolumny dziesięciu znaków kreator wpisze do kolejnej następne osiem i tak dalej.

Nasz plik jest rozdzielany średnikiem. Zdarza się, że dane są w pliku poprzedzone nagłówkiem. System generujący plik tekstowy może umieścić w nim datę wygenerowania pliku, swoją nazwę lub inny komentarz do danych. Pole *Rozpocznij import od wiersza* pozwala nam ominąć te informacje i zacząć pobieranie danych od odpowiedniego miejsca. W naszym pliku `szkoly.txt` w pierwszym wierszu są nagłówki kolumn, znowu więc zostawiamy domyślne ustawienia.

Pole *Pochodzenie pliku* najczęściej zostawiamy bez zmian. Jeżeli w podglądzie pliku poniżej zamiast liter widać "krzaczki", niezrozumiałe znaki, trzeba zmienić tę opcję.

Krok drugi wygląda różnie, zależnie od tego, jaki rodzaj pliku wybraliśmy przed chwilą.

Krok 2 z 3 — plik rozdzielany

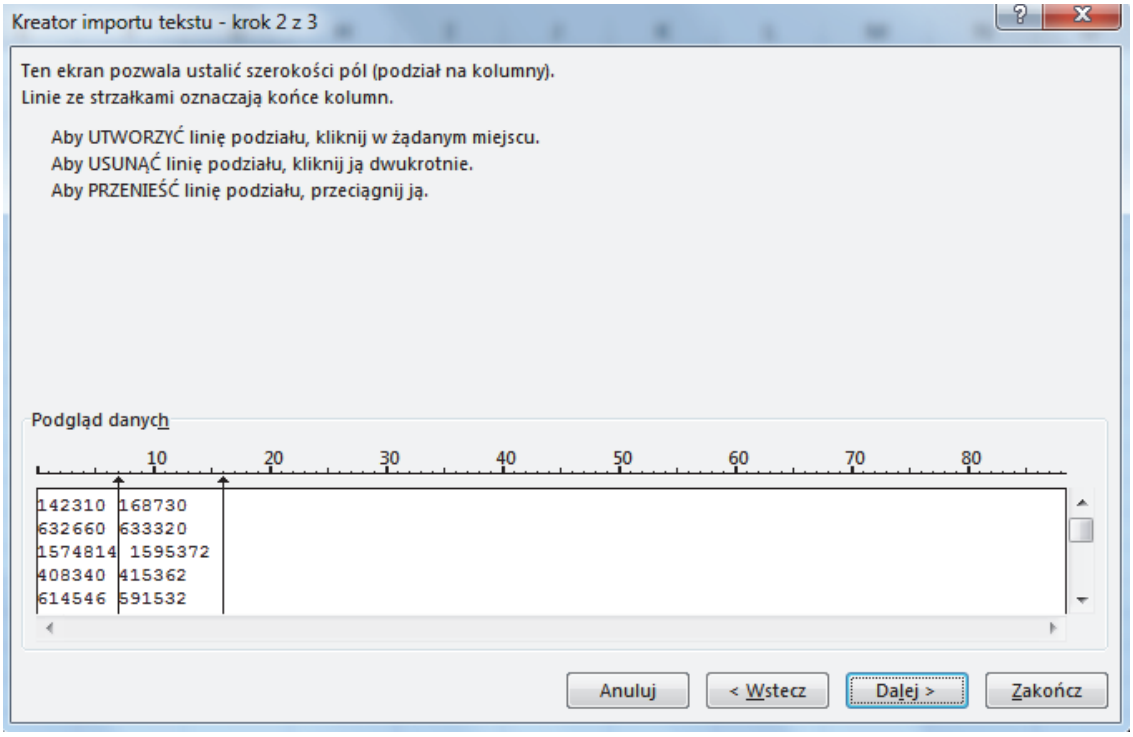


Rysunek 5 — Krok drugi — dla pliku rozdzielanego

W tym kroku określamy, jaki znak rozdziela kolumny — w naszym pliku to średnik. Poniżej widać podgląd importowanego pliku, odpowiednie dane są już w swoich kolumnach.

Krok 2 z 3 — stała szerokość

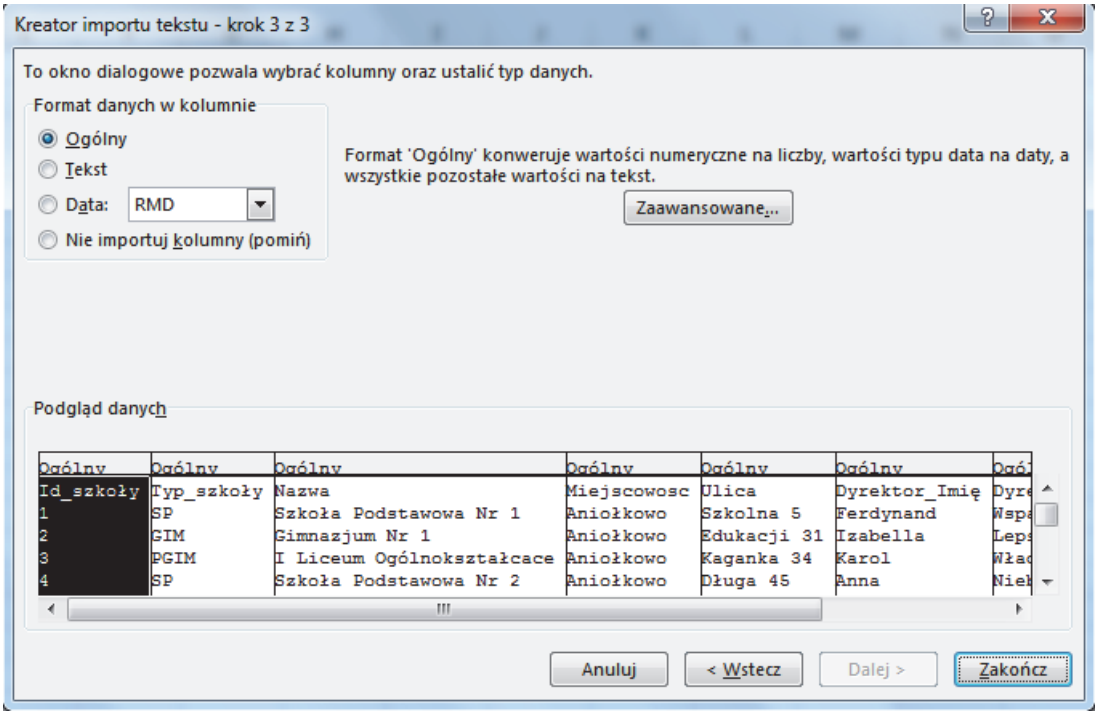
Nie dotyczy to naszego pliku, jednak spójrzmy na chwilę, jak wygląda krok drugi w tym przypadku, oczywiście dla innego pliku.



Rysunek 6 — Krok drugi dla pliku o stałej szerokości pól

W powyższym oknie musimy wskazać, gdzie zaczynają się kolejne kolumny danych, zaznaczając kursorem miejsce podziału.

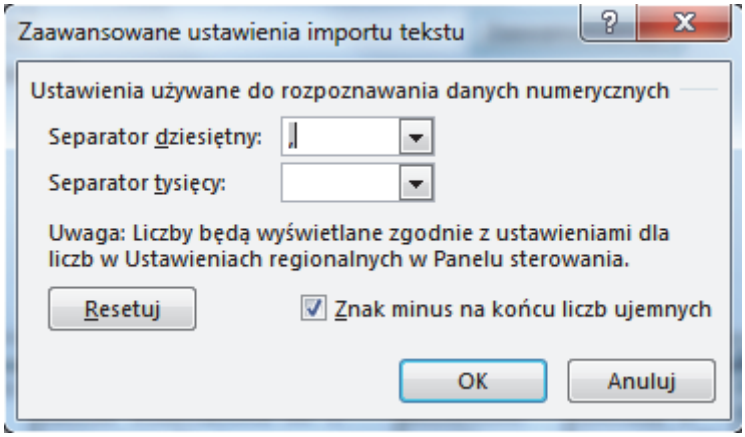
Krok 3 z 3



Rysunek 7 — Krok trzeci importu

W trzecim kroku na ogół akceptujemy domyślne ustawienia kreatora importu. Zwróćmy jednak uwagę na dostępne tu opcje. Wybierając kolumny w podglądzie danych, możemy zmieniać sposób, w jaki są one importowane. Możemy też zdecydować, które kolumny powinny być pominięte w procesie importu. Szczególnie należy zwrócić uwagę na przycisk *Zaawansowane*.

W tym miejscu potrzebna jest mała dygresja. W polskim systemie Windows część całkowita i ułamkowa oddzielane są przecinkiem. Ustawienia stosowane przez system Windows można sprawdzić w opcji *Ustawienia regionalne* Panelu sterowania. Plik tekstowy może pochodzić z innego systemu, w którym na przykład do tego stosowana jest.



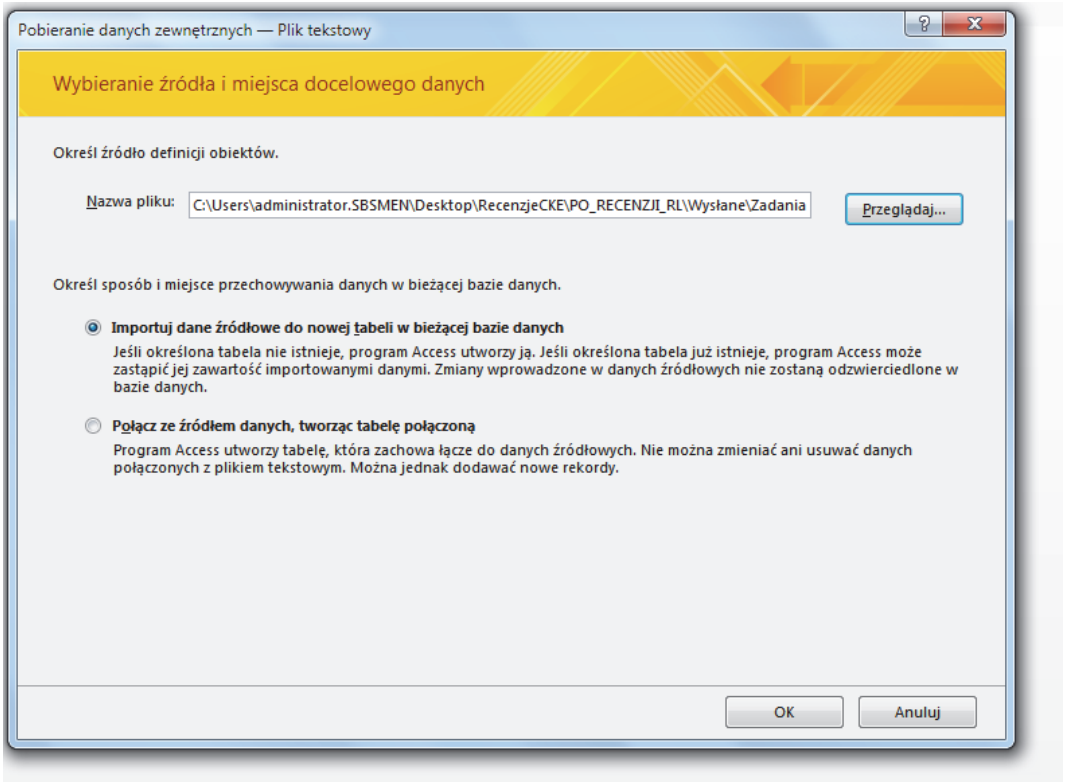
Rysunek 8 — Ustawienia zaawansowane importu liczb

Na przykład liczba 1982,45 zostanie zinterpretowana w polskim systemie prawidłowo, ale ta liczba może być też zapisana jako 1,982.45 (w zapisie amerykańskim). Przy poniższych ustawieniach podpowiadamy, jak zapisać liczby w naszym pliku w taki sposób, aby w jego komórkach ostatecznie pojawiły się właściwe liczby. Dzięki temu będziemy mogli wykonywać różne działania (sumy, średnie itp.). Przy błędnych ustawieniach Excel zaimportuje ciąg znaków „1,982.45", będzie go traktował jako tekst i nie pozwoli nam na działania, jakie byłyby dostępne w przypadku liczb.

W przypadku naszego pliku możemy jednak zostawić ustawienia domyślne. Naciskam więc *Zakończ* i następnie *OK*, aby zaobserwować, jak mój arkusz wypełnia się danymi z pliku.

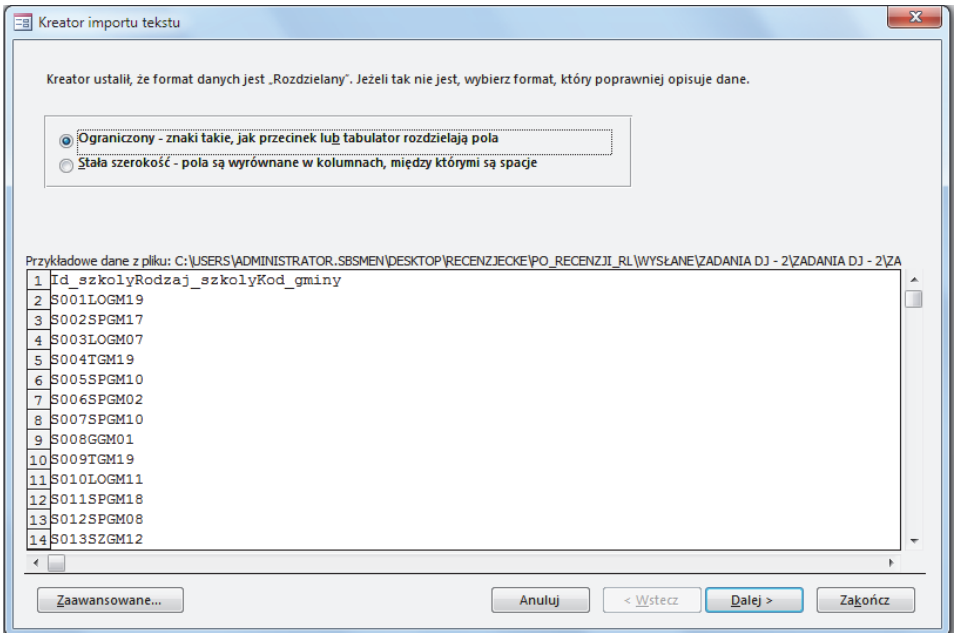
Importowanie danych do MS Access

Dane importujemy do założonej pustej bazy danych, przy czym dla każdego pliku tworzymy osobną tabelę. Zademonstrujmy import na przykładzie pliku `szkoly.txt`, który zawiera zestawienie różnych szkół w poszczególnych gminach. Pierwszy wiersz jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.



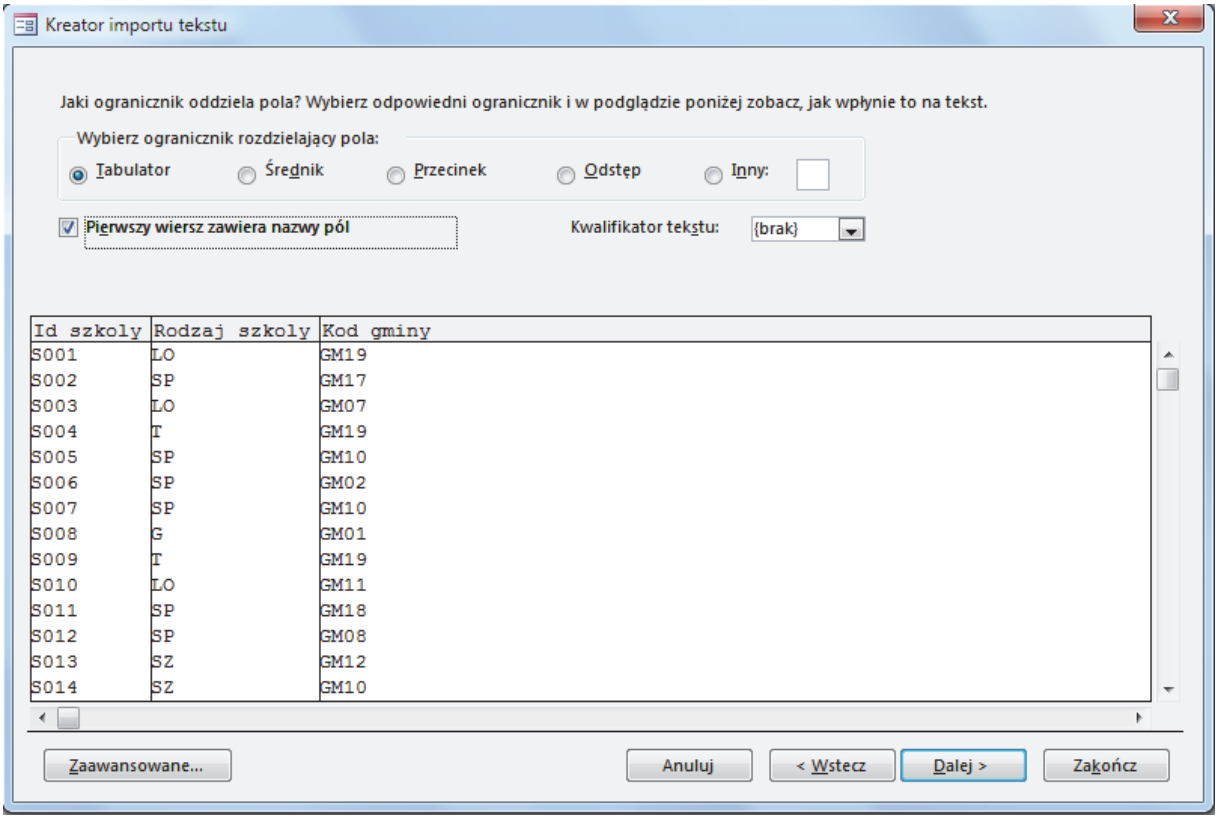
Krok 1 to wywołanie kreatora importu:

Decydujemy, jaki jest format danych.

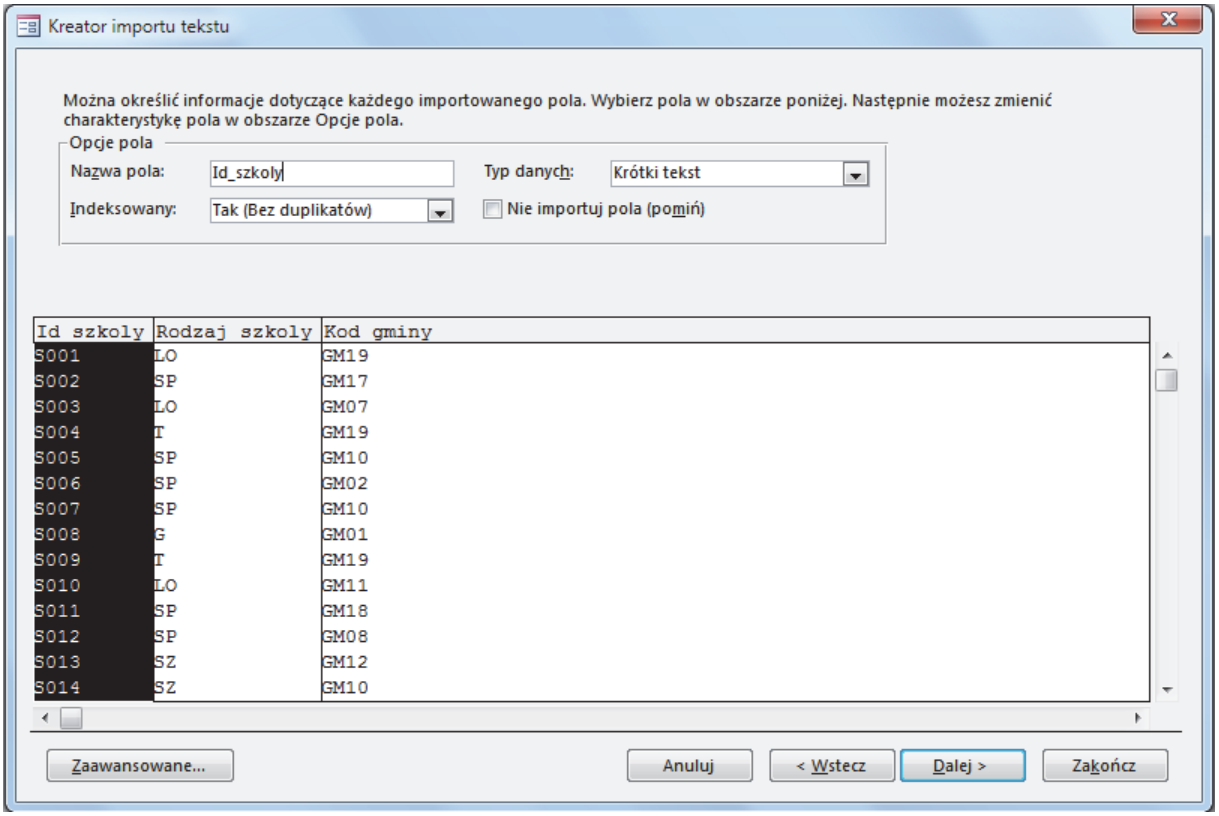


Wybieramy w tym wypadku format ograniczony separatorem (to może być znak tabulacji, przecinek, średnik, odstęp lub inny znak).

W **kroku 2** zaznaczamy, który znak jest separatorem oddzielającym pola, oraz zaznaczamy (lub nie), czy pierwszy wiersz zawiera nazwy pól:



Następnie określamy informacje dotyczące **każdego pola**: jego nazwę, typ przechowywanych danych, to, czy pole będzie indeksowane i czy może zawierać duplikaty.



Nie zawsze wybór typu danych dla pola jest oczywisty na pierwszy rzut oka. Przykład: importując kolumnę zawierającą numery PESEL, wybieramy dla pola typ tekstowy, ponieważ wartości liczbowe numerów PESEL są zbyt duże, aby przechowywać je w polu typu *liczba całkowita* (a nawet *liczba całkowita długa*).

Problem może też stwarzać import danych zawierających zera wiodące. Jeśli zaimportujemy takie dane do pola typu liczbowego, to zera wiodące zostaną usunięte, na przykład dana 0020 po zaimportowaniu będzie mieć wartość 20. Aby utrzymać zera wiodące, taką daną importujemy do pola typu tekstowego.

Kolejna decyzja dotyczy klucza głównego tabeli, a ściślej: czy ma on w ogóle być, a jeżeli tak, to jaki. MS Access domyślnie proponuje dodanie kolumny stanowiącej klucz główny, warto jednak samodzielnie zdecydować, które z już istniejących pól może pełnić rolę klucza. Możemy także zdecydować, że nasza tabela nie będzie miała klucza podstawowego.

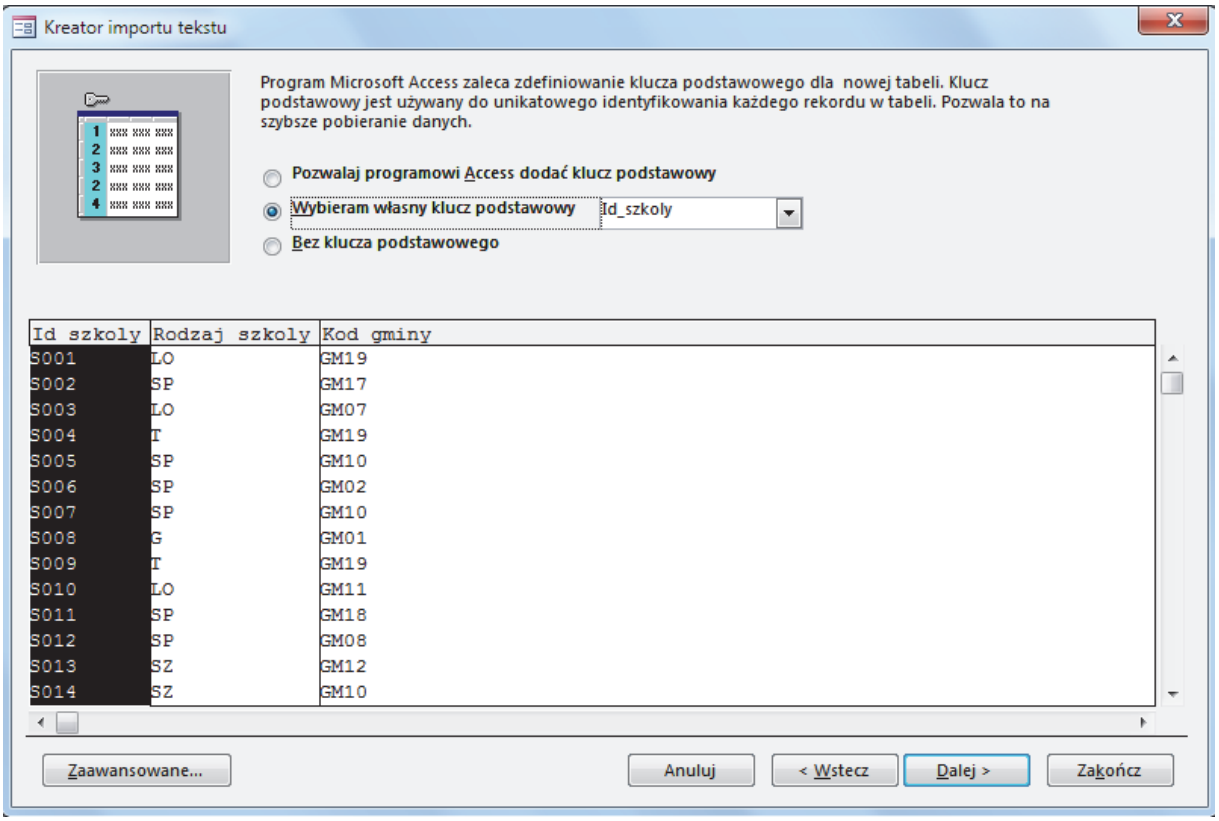
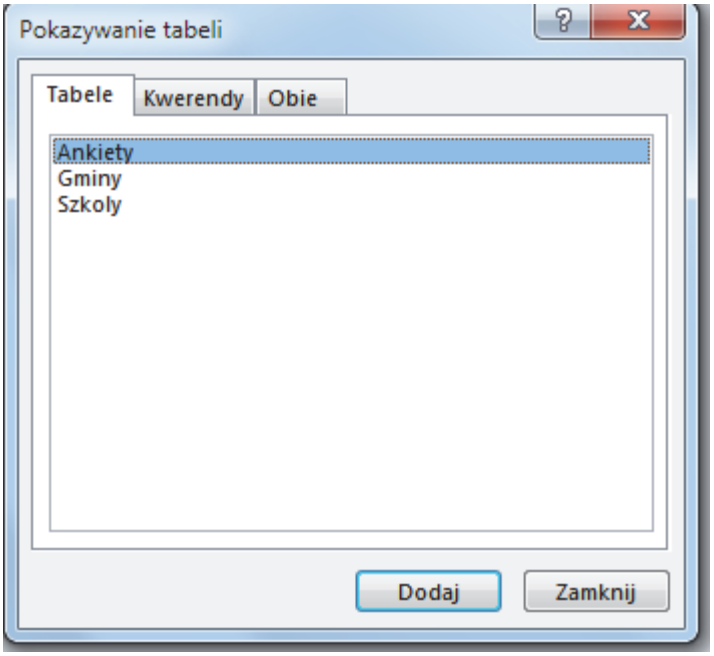


Tabela domyślnie otrzymuje nazwę taką jak nazwa pliku, z którego pochodzą dane, ale możemy tę nazwę zmienić.

W programie Microsoft Access można po zaimportowaniu danych utworzyć wiązania wybranych pól z różnych tabel i zapisać je jako stały element systemu bazodanowego (relacje). Można również tworzyć wiązania tylko na potrzeby konkretnego zapytania (kwerendy) — jest to naturalny sposób programowania w języku SQL.

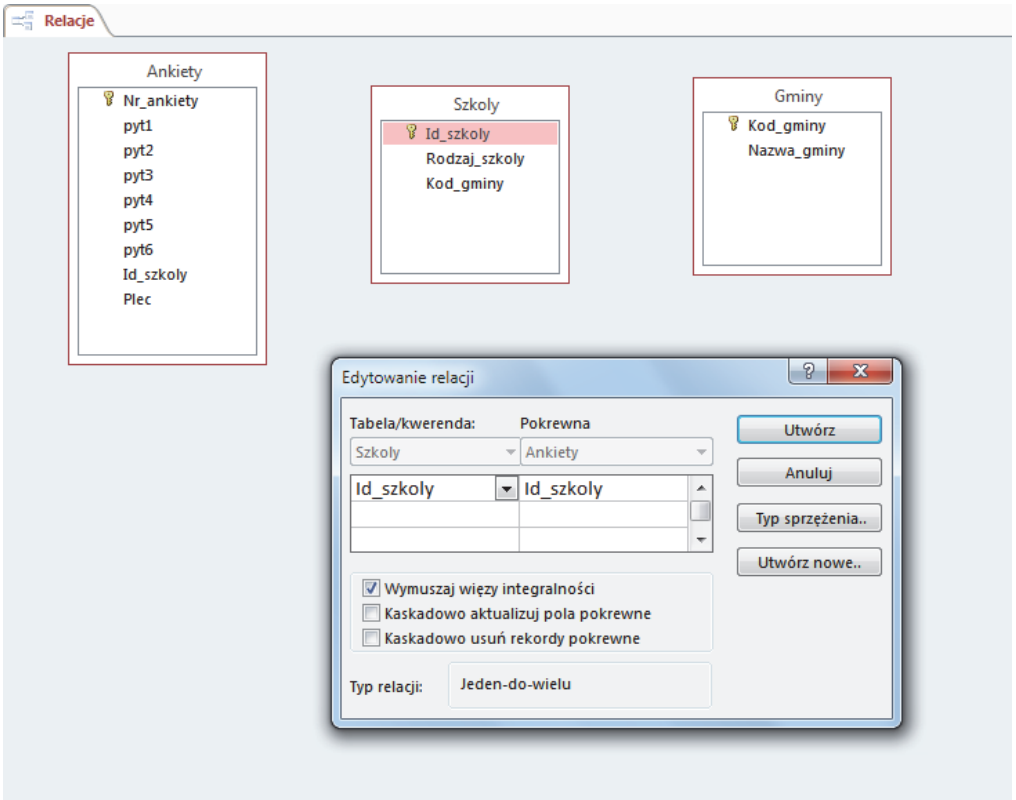
Jeśli zdecydujemy się utworzyć relacje na początku, w programie Microsoft Access wybieramy zakładkę Narzędzia bazy danych/Relacje. Wybieramy te spośród tabel, dla których chcemy stworzyć relacje, czyli na ogół wszystkie tabele w bazie.

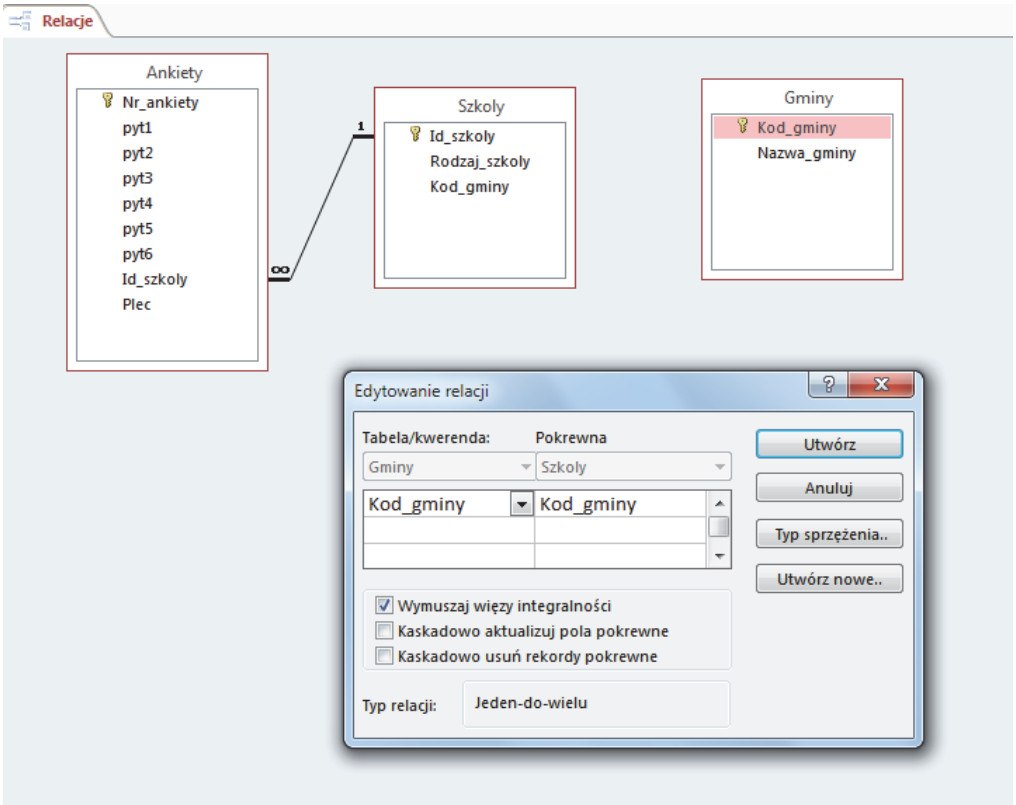


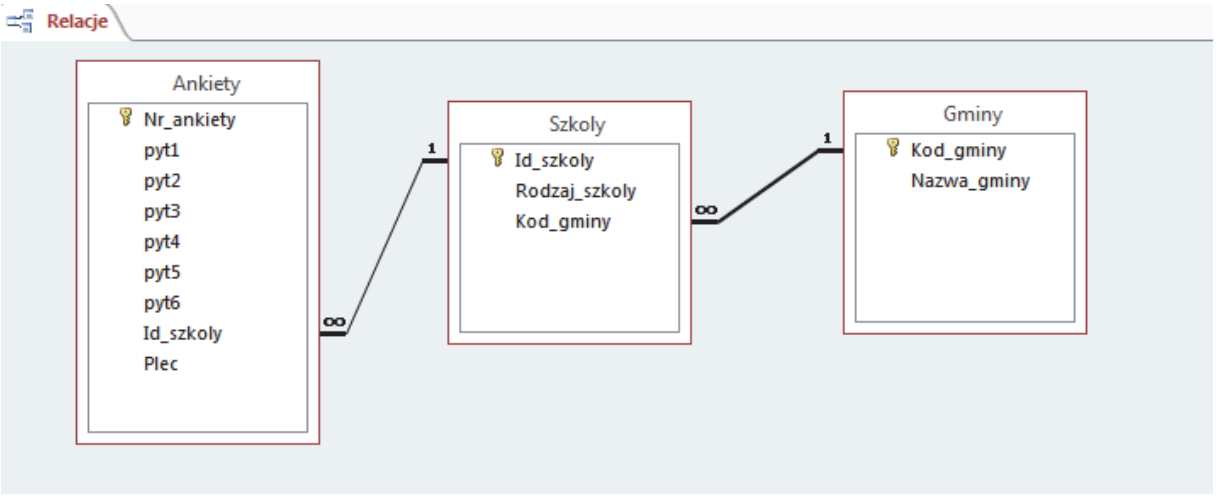


Wiązanie tworzymy, przeciągając myszką wybrane pole z jednej tabeli i „puszczając" je nad wybranym polem drugiej tabeli. Oba pola muszą być tego samego typu, choć mogą mieć różne nazwy. Domyślnie tworzone jest wiązanie typu *wewnętrzne*, które z obu wiązanych tabel udostępnia tylko rekordy pasujące. We właściwościach wiązania można zmienić jego typ na zewnętrzne lewo- lub prawostronne, które będzie udostępniać wszystkie rekordy z jednej tabeli, a z drugiej tabeli tylko rekordy pasujące do połączenia.

W oknie kreatora, obok linii łączącej pola, pojawia się informacja o typie relacji: 1:1 lub 1:∞.







Wczytywanie i wypisywanie danych w zadaniach programistycznych

W języku C++ do odczytu i zapisu danych służą specjalne obiekty zwane *strumieniami*. Taki obiekt powiązany jest ze źródłem danych — w naszym wypadku będą to zwykle pliki na dysku, ale można też obsługiwać strumienie odpowiadające np. urządzeniom zewnętrznym.

Aby użyć w swoim programie strumieni powiązanych z plikami, należy zadeklarować użycie biblioteki *fstream* (z angielskiego *„file-stream”*, czyli „strumienie plikowe”):

```
#include <fstream>
```

W samym programie deklarujemy obiekt, który będzie czytał z konkretnego pliku. Obiekt taki powinien mieć jeden z następujących trzech typów:

- *ifstream* — strumień wejścowy, czyli służący do czytania danych z pliku (*input file stream*)

- *ofstream* — strumień wyjściowy, do zapisywania danych do pliku (*output file stream*)
- *fstream* — strumień, z którego można zarówno czytać dane, jak i zapisywać dane do niego.

Deklaracja obiektu wygląda tak, jak każdej innej zmiennej w C++. Wybierzmy dla niego nazwę „wejscie”:

```
ifstream wejscie;
```

Teraz kojarzymy program z konkretnym plikiem na dysku — niech nazywa się, na przykład, *dane.txt:*

```
wejscie.open("dane.txt");
```

Od tej chwili zmienna strumieniowa *wejscie* pobiera kolejne dane z pliku dane.txt. Nadmieńmy jeszcze, że te dwie instrukcje można połączyć w jedną, mającą ten sam skutek:

```
ifstream wejscie("dane.txt");
```

Ten obiekt jest typu *ifstream*, czyli służy do czytania danych. Deklaracja obiektu, który odpowiada za zapis danych do pliku wygląda bardzo podobnie:

```
ofstream wyjscie;
wyjscie.open("wynik.txt");
```

albo:

```
ofstream wyjscie("wynik.txt");
```

Wreszcie, zmienna typu *fstream* może służyć zarówno do czytania, jak i do pisania, trzeba jednak przy otwieraniu pliku podać, do czego w tym momencie zostanie użyta, za pomocą następującej konstrukcji:

```
fstream plik;
plik.open("plik.txt", ios::in);
```

jeśli zamierzamy czytać z pliku, lub:

```
plik.open("plik.txt", ios::out);
```

jeśli będziemy do niego pisać.

Ważne: deklaracja zmiennej typu *ofstream*, albo *fstream* z parametrem *ios::out* spowodują, że **plik zostanie wyczyszczony**, a nowe dane zastąpią to, co było w nim wcześniej. Aby dopisać nowe dane na koniec istniejącego pliku, trzeba użyć instrukcji:

```
ofstream plik;
plik.open("wynik.txt",ios::app);
```

Oczywiście można użyć wariantu z pojedynczą instrukcją:

```
ofstream plik("wynik.txt",ios::app);
```

albo użyć *fstream* w miejsce *ofstream*.

Mamy zatem pliki otwarte do odczytu i zapisu — w zadaniach maturalnych najczęściej będziemy potrzebowali jednego pliku do odczytu i jednego do zapisu.

Dane czytamy ze strumieni i piszemy do strumieni za pomocą operatorów << i >>. Jeśli, na przykład, mamy zadeklarowaną zmienną typu *int*:

```
int a;
```

i chcemy odczytać jedną liczbę z pliku, któremu odpowiada zmienna strumieniowa *wejscie*, piszemy:

```
wejscie >> a;
```

Jeśli chcemy odczytać dwie liczby i zapisać je do zmiennych *a* i *b*, możemy napisać:

```
wejscie >> a >> b;
```

co jest skrótem dla dwóch osobnych instrukcji (`wejscie >> a; wejscie >> b;`)

Jeśli, na przykład, w pliku jest 1000 liczb, zapisanych każda w osobnym wierszu, możemy zadeklarować tablicę odpowiedniej długości:

```
int A[1000];
```

a następnie przeczytać wszystkie liczby za pomocą pętli for:

```
for(int i=0; i<1000; i++)
     wejscie >> A[i];
```

Dla przykładu, analogicznie do pliku wyjściowego zapisujemy np. pojedynczą liczbę za pomocą instrukcji:

```
wyjscie << a;
```

Rozwiążmy dla przykładu bardzo proste zadanie: w pliku `dane.txt` znajduje się 1000 wierszy, z których każdy zawiera po dwie liczby, oddzielone pojedynczym odstępem. Należy dla każdej pary liczb policzyć ich sumę i zapisać ją do pliku `wynik.txt`, każdą sumę w osobnym wierszu. Odpowiedni program może wyglądać tak:

```
#include <fstream>

using namespace std;
     // zaznaczamy której przestrzeni nazw będziemy używać

int main()
{
    // deklarujemy tablice A i B długości 4
    int A[4], B[4];

    // zmienna strumieniowa na plik wejsciowy
    ifstream wejscie("dane.txt");

    // zmienna strumieniowa na plik wejsciowy
    ofstream wyjscie("wynik.txt");


    for(int i=0; i<4; i++)
        // wczytaj z pliku dwie liczby i zapisz je do A[i] oraz B[i]
        wejscie >> A[i] >> B[i];


    for(int i=0; i<4; i++)
        // wypisz sumę do pliku wyjściowego
        wyjscie << A[i]+B[i] << endl;


    wejscie.close();
```

```
    wyjscie.close();
}
```

W tym programie użyliśmy dwóch dodatkowych instrukcji. Instrukcja

```
wyjscie << endl;
```

wypisuje do pliku znak końca wiersza (czyli przechodzi do następnego wiersza). Instrukcje

```
wejscie.close();
wyjscie.close();
```

zamykają pliki, czyli informują system operacyjny, że nie będziemy już ich potrzebować. Brak instrukcji zamknięcia pliku może spowodować (choć na ogół nie powoduje) różnego rodzaju problemy (np. niekompletne zapisanie się danych).

Warto zauważyć, że w tym programie nie musieliśmy najpierw czytać wszystkich liczb z pliku wejściowego, a potem wypisywać wszystkich sum. Jak najbardziej możliwe jest wczytanie dwóch liczb, wypisanie ich sumy, a potem zajęcie się kolejnymi dwiema liczbami. W takich sytuacjach nie potrzebujemy nawet tablic do przechowywania danych, co skraca program:

```
#include <fstream>

using namespace std;

int main()
{
    int a, b;
    ifstream wejscie("dane.txt");
    ofstream wyjscie("wynik.txt");
    for(int i=0; i<4; i++)
    {
        // wczytaj z pliku dwie liczby
        wejscie >> a >> b;
        // wypisz sumę do pliku wyjściowego
        wyjscie << a+b << endl;
    }
    wejscie.close();
    wyjscie.close();
}
```

Czasem zamiast liczb chcemy czytać napisy (łańcuchy znaków), które w C++ przechowuje się w zmiennych typu *string*. Wczytywanie i wypisywanie takich danych wygląda prawie identycznie:

```
string S;       // deklaracja zmiennej typu łańcuchowego
wejscie >> S;
```

Warto wiedzieć, że taka instrukcja wczytuje znaki tylko **tak długo, aż napotka pierwszy biały znak** (spację, koniec wiersza, tabulatora). Jeśli w pliku w pierwszym wierszu znajduje się napis:

```
Dzien dobry!
```

to instrukcja

```
wejscie >> S;
```

zapisze do zmiennej S tylko słowo „Dzien”. Gdybyśmy mieli dwie zmienne, S i T, obie typu *string*, to instrukcja:

```
wejscie >> S >> T;
```

zapisałaby do S napis „Dzien”, a do T napis „dobry!”, ignorując spacje.

Jednak przy wyprowadzaniu tej danej do pliku wyjściowego, zapisany zostanie zawsze cały łańcuch. Instrukcja:

```
S = ”Dzien dobry!”
wyjscie << S;
```

umieści w pliku wyjściowym pełny napis „Dzien dobry!”.

Aby więc na przykład przeczytać z pliku wejściowego 200 słów, z których każde zapisane jest w osobnym wierszu, należy wykonać następujący fragment programu:

```
string Tablica[200];
for(int i=0; i<200; i++)
  wejscie >> Tablica[i];
```

Tego samego programu można użyć, jeśli słowa nie są zapisane w osobnych wierszach, ale oddzielone odstępami.

Wczytywanie/wypisywanie liczb i napisów można bez problemu mieszać ze sobą, na przykład instrukcja:

```
string a = ”Basnie z ”;
int b = 1000;
string c = ” i jednej nocy”;

wyjscie << a << b << c;
```

wypisze `Basnie z 1000 i jednej nocy`. Zawartość np. takiego pliku:

```
Zakroczym 2
Wyszogrod 34
Plock 71
Wloclawek 115
```

zostanie prawidłowo odczytana, jeśli wykonamy następujące instrukcje:

```
string A[4];
int B[4];

for(int i=0; i<4; i++)
        wejscie >> A[i] >> B[i];
```

Napisy (nazwy miast) znajdą się w tablicy A, zaś liczby (odległości) — w tablicy B.

Czasem zadanie wymaga pracy na liczbach, które nie są całkowite — w C++ do przechowywania takich liczb służą zmienne typu *float* i *double*. Jeśli na przykład zmiennej *x* przypiszemy wartość 1/7:

```
double x = 1.0 / 7;
```

i użyjemy polecenia

```
wyjscie << x;
```

do pliku wyjściowego zostanie wypisane `0.142857`. W zadaniach maturalnych potrzebujemy jednak najczęściej pewnej ustalonej precyzji zapisu dziesiętnego — na przykład polecenie może brzmieć: „Wypisz liczbę z dokładnością do dwóch miejsc po przecinku”. Aby ustalić precyzję, posługujemy się instrukcją *precision*, używaną w następujący sposób:

```
wyjscie.precision(3);
wyjscie << x;
```

Teraz otrzymujemy wynik 0.143, jednak wypisanie tą samą metodą 1/70 może prowadzić do niespodzianek — fragment kodu:

```
double x = 1.0 / 70;
wyjscie.precision(3);
wyjscie << x;
```

spowoduje wypisanie `0.0143`, a nie `0.014`, ponieważ kompilator, wypisuje trzy cyfry **znaczące**, czyli nie liczy zer bezpośrednio po przecinku. Aby wypisywał on zawsze dokładnie trzy cyfry po przecinku, konieczna jest jeszcze instrukcja *fixed*:

```
wyjscie.precision(3);
wyjscie << fixed;
wyjscie << x;
```

Aby używać instrukcji *precision* i *fixed*, konieczne jest dołączenie nagłówka *iomanip* na początku programu:

```
#include <iomanip>
```

Warto pamiętać, że jeśli nie zmieniamy precyzji, domyślnie użytym zapisem w C++ jest 6 cyfr znaczących.

2.1. Algorytmy w praktyce

Zadanie 58.

Wiązka zadań *Systemy liczbowe*

Centralny ośrodek meteorologiczny planety Cyfrak codziennie w południe rejestruje wskazania zegarów oraz temperaturę w trzech stacjach pogodowych: S1, S2, S3. Zegary w stacjach pogodowych odliczają liczbę godzin, które upłynęły od uruchomienia stacji. W stacji S1 wszystkie wartości (wskazania zegara i temperatury) zapisywane są w systemie binarnym, w stacji S2 — w systemie czwórkowym (czyli systemie pozycyjnym o podstawie 4), a w stacji S3 — w systemie ósemkowym (czyli systemie pozycyjnym o podstawie 8). Temperatury ujemne poprzedzone są znakiem „–”, np. –1101 w systemie dwójkowym oznacza liczbę o zapisie dziesiętnym –13.

Pliki `dane_systemy1.txt`, `dane_systemy2.txt`, `dane_systemy3.txt` zawierają wyniki 1095 kolejnych pomiarów przeprowadzonych w stacjach S1, S2, S3 od czasu ich uruchomienia. Każdy wiersz pliku zawiera wyniki jednego pomiaru: stan zegara i temperaturę. Wartości w wierszach rozdzielone są spacjami.

Przykład

Wiersz opisujący pomiar, w którym zegar wskazuje liczbę 36, a termometr temperaturę –7, wyglądałby następująco:

Plik `dane_systemy1.txt`
```
100100 -111
```
Plik `dane_systemy2.txt`
```
210 -13
```
Plik `dane_systemy3.txt`
```
44 -7
```

Napisz program(-y), który pozwoli rozwiązać poniższe zadania. Odpowiedzi zapisz w pliku `wyniki_systemy.txt`. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją numerem zadania.

58.1.

Dla każdej stacji pogodowej podaj najniższą zarejestrowaną temperaturę, a wszystkie wyniki zapisz w systemie binarnym (dwójkowym).

58.2.

Zgodnie z harmonogramem pomiary wykonywane są co 24 godziny, począwszy od pierwszego pomiaru. Oznacza to, że wyrażone dziesiętnie stany zegarów w kolejnych pomiarach powinny wynosić 12, 12+24=36, 12+2·24=60 itd.

Podaj liczbę pomiarów, w których zarejestrowany stan zegara był niepoprawny jednocześnie we wszystkich stacjach pogodowych.

Przykład

Rozważmy dane, w których pierwsze 3 wiersze pliku ***dane_systemy1.txt*** są następujące:

```
1100 -11
```

```
100100 -111
111101 1
```

Ponieważ zapisane binarnie stany zegara: `1100`, `100100 i 111101` to odpowiednio wartości: 12, 36 i 61, to tylko stany podane w dwóch pierwszych wierszach są poprawne, zaś w trzecim wierszu stan jest nieprawidłowy.

58.3.

Rekordem temperatury dla danej stacji pogodowej nazywać będziemy pomiar temperatury, który jest większy od wszystkich wcześniejszych pomiarów dokonanych w tej stacji.

Przykład

Dla następujących wyników kolejnych pomiarów temperatur dokonanych od pierwszego pomiaru w danej stacji (podanych w zapisie dziesiętnym):

1, -1, 0, 2, 1, 1, 3, 4, 4, 3, 1, 7, 2, 1

rekordami temperatury są wszystkie podkreślone wyniki.

Dniem rekordowym jest dzień, w którym **w co najmniej jednej** stacji pogodowej zarejestrowano rekord temperatury. Podaj liczbę dni rekordowych.

Przykład: przyjmijmy, że — podane w zapisie dziesiętnym — wyniki pomiarów dokonywanych w kolejnych dniach były w trzech stacjach następujące:

Dzień	**S_1**	**S_2**	**S_3**
1	**1**	**0**	**-1**
2	**2**	**1**	-1
3	1	-1	-1
4	0	-2	0
5	1	**2**	**1**

Dla powyższych danych liczba dni rekordowych wynosi: **3**.

58.4.

Oznaczmy kolejne zarejestrowane temperatury w stacji pogodowej S1 przez t_1, t_2, t_3,… Niech r_{ij} oznacza kwadrat różnicy między temperaturami w i-tym i j-tym pomiarze pierwszej stacji pogodowej, $r_{ij}=(t_i - t_j)^2$. *Skokiem temperatury* między i-tym a j-tym pomiarem nazywać będziemy zaokrąglenie w górę do liczby całkowitej ułamka $r_{ij} / |i - j|$.

Przykład

Dla następujących kolejnych pomiarów temperatur (zapisanych dziesiętnie):

3, 5, 4, 7,

skoki temperatur opisuje poniższa tabela

i, j	***t_i, t_j***	***r_{ij}***	***$\|i - j\|$***	**Skok temperatury między *i*-tym a *j*-tym pomiarem**
1, 2	3, 5	$2^2=4$	1	4
1, 3	3, 4	$1^2=1$	2	1
1, 4	3, 7	$4^2=16$	3	6
2, 3	5, 4	$1^2=1$	1	1

2, 4	5, 7	$2^2=4$	2	2
3, 4	4, 7	$3^2=9$	1	9

Podaj największy skok temperatury w stacji pogodowej S1. Wynik podaj w systemie dziesiętnym.

Zadanie 59.

Wiązka zadań *Ciekawe liczby*

W pliku `liczby.txt` w oddzielnych wierszach znajduje się **1000 różnych liczb**, każda o długości od 2 do 9 cyfr. **Napisz program(-y)**, który da odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku `wyniki_liczby.txt`, a każdą odpowiedź poprzedź numerem zadania.

59.1.

Czynnikiem pierwszym danej liczby naturalnej złożonej jest dowolna liczba pierwsza, która dzieli tę liczbę całkowicie. Podaj, ile jest w pliku `liczby.txt` liczb, w których rozkładzie

na czynniki pierwsze występują **dokładnie trzy różne czynniki** (mogą się one powtarzać, z których każdy jest **nieparzysty**.

Przykład

Liczba	**Czynniki pierwsze**	**Czy w rozkładzie występują dokładnie trzy różne nieparzyste czynniki pierwsze?**
32	2, 2, 2, 2, 2	NIE
210	2, 3, 5, 7	NIE
1331	11, 11, 11	NIE
1157625	3, 3, 3, 5, 5, 5, 7, 7, 7	TAK
105	3, 5, 7	TAK
429	3, 11, 13	TAK
1287	3, 3, 11, 13	TAK
3465	3, 3, 5, 7, 11	NIE
255255	3, 5, 7, 11, 13, 17	NIE

59.2.

Podaj, ile jest w pliku `liczby.txt` liczb, dla których suma danej liczby i liczby odwróconej jest liczbą palindromiczną, tzn. jej zapis dziesiętny jest palindromem.

Przykład

Liczba	Liczba odwrócona	Suma	Czy suma jest palindromem?
45	54	99	TAK
471046105	501640174	972686279	TAK
11264	46211	57475	TAK
19	91	110	NIE
8542	2458	11000	NIE

59.3.

Niech $w(n)$ oznacza iloczyn cyfr liczby n. Dla danej liczby n tworzymy ciąg, w którym kolejny element jest iloczynem cyfr występujących w poprzednim elemencie:

$$n_1 = w(n)$$
$$n_2 = w(n_1)$$
$$n_3 = w(n_2)$$
$$\ldots$$

Ciąg kończy się, gdy liczba n_k jest liczbą jednocyfrową. Wówczas **mocą liczby *n*** jest liczba k.

Podaj, ile jest w pliku `liczby.txt` liczb o mocy 1, 2, 3, ..., 8. Dodatkowo podaj minimalną i maksymalną liczbę o mocy równej 1.

Przykład

Liczba 678 ma moc **4**, ponieważ:

$$6 * 7 * 8 = 336$$

$$3 * 3 * 6 = 54$$

$$5 * 4 = 20$$

$$2 * 0 = 0$$

Liczba 1991 ma moc **2**, ponieważ

$$1 * 9 * 9 * 1 = 81$$

$$8 * 1 = 8$$

Zadanie 60.
Wiązka zadań *Dzielniki*

W pliku `liczby.txt` danych jest 200 różnych liczb całkowitych z przedziału [2, 1 000 000], każda w osobnym wierszu pliku. Napisz program (lub kilka programów), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku `wyniki.txt`.

60.1.

Policz, ile jest w pliku wejściowym liczb mniejszych niż 1000, oraz podaj dwie takie liczby, które pojawiają się w pliku jako ostatnie (możesz założyć, że będą co najmniej dwie).

60.2.

Wśród liczb występujących w pliku wejściowym znajdź te, które mają dokładnie 18 dzielników naturalnych (wliczając w nie 1 i samą liczbę). Dla każdej znalezionej liczby wypisz, oprócz jej wartości, listę wszystkich jej dzielników, posortowaną rosnąco.

60.3.

Znajdź największą liczbę w pliku, która jest względnie pierwsza ze wszystkimi pozostałymi, czyli taką, która z żadną z pozostałych liczb nie ma wspólnego dzielnika innego niż 1.

Zadanie 61.

Wiązka zadań *Ciągi arytmetyczne*

Ciąg liczb całkowitych nazywamy *ciągiem arytmetycznym*, jeśli różnica między każdymi dwoma kolejnymi jego wyrazami jest identyczna. Ciągiem arytmetycznym jest na przykład ciąg (1, 3, 5, 7, 9). Jest to ciąg o różnicy 2, ponieważ każdy wyraz tego ciągu, poza pierwszym, różni się od poprzedniego wyrazu o 2. Ciąg (17, 22, 27, 32, 37) jest ciągiem arytmetycznym o różnicy 5. W tym zadaniu rozpatrujemy tylko takie ciągi arytmetyczne, które mają dodatnią różnicę oraz co najmniej pięć wyrazów.

W pliku *ciagi.txt* danych jest 100 ciągów składających się z liczb całkowitych dodatnich, nieprzekraczających 1 000 000. Każdy ciąg opisany jest w dwóch wierszach: pierwszy zawiera liczbę wyrazów ciągu (co najmniej 5 i co najwyżej 1000), zaś drugi — kolejne wyrazy ciągu, oddzielone pojedynczymi odstępami. Dla przykładu pierwsze cztery wiersze pliku mają następującą postać:

```
5
1 3 6 7 9
5
17 22 27 32 37
```

Napisz program (lub kilka programów), który wykona podane poniżej polecenia.

61.1.

Podaj, ile spośród podanych w pliku `ciagi.txt` ciągów jest ciągami arytmetycznymi. Znajdź wśród nich ciąg o największej różnicy i oblicz jego różnicę. Liczbę ciągów arytmetycznych oraz największą różnicę zapisz w pliku `wynik1.txt`.

61.2.

Dla każdego podanego ciągu znajdź — jeśli istnieje — największą występującą w nim liczbę, która jest pełnym sześcianem jakiejś liczby naturalnej (w pierwszym z przykładowych ciągów jest to $1 = 1^3$, w drugim — $27 = 3^3$).

Znalezione liczby (po jednej dla każdego ciągu, w którym taka liczba występuje) zapisz w pliku `wynik2.txt`, w kolejności zgodnej z kolejnością ciągów, z których pochodzą.

61.3.

Plik `bledne.txt` ma identyczną strukturę jak `ciagi.txt`, ale zawiera tylko 20 ciągów. Wiadomo jednak, że wszystkie występujące w nim ciągi są ciągami arytmetycznymi z jednym błędem: jeden z wyrazów w każdym ciągu został zastąpiony przez liczbę naturalną nienależącą do ciągu.

Dla każdego ciągu znajdź i wypisz błędny wyraz. Odpowiedzi zapisz w pliku `wynik3.txt`, podając dla każdego ciągu błędną liczbę w osobnym wierszu, w kolejności zgodnej z kolejnością ciągów w pliku wejściowym.

Zadanie 62.

Wiązka zadań *Liczby ósemkowe*

W pliku `liczby1.txt` znajduje się 1000 liczb całkowitych dodatnich, zapisanych ósemkowo, maksymalnie sześciocyfrowych. Każda liczba umieszczona jest w osobnym wierszu.

W pliku `liczby2.txt` znajduje się 1000 liczb całkowitych dodatnich, zapisanych dziesiętnie, maksymalnie sześciocyfrowych. Każda liczba umieszczona jest w osobnym wierszu.

Napisz program(-y), za pomocą którego(-ych) rozwiążesz poniższe zadania. Do oceny oddaj dokument `wyniki.txt` z rozwiązaniami poszczególnych zadań oraz pliki źródłowe programów wykorzystanych do uzyskania rozwiązania.

62.1.

Wyszukaj w pliku `liczby1.txt` dwie liczby, najmniejszą i największą. Podaj wartości tych liczb w zapisie ósemkowym.

62.2.

Znajdź najdłuższy niemalejący ciąg liczb występujących w kolejnych wierszach pliku `liczby2.txt`. Podaj pierwszy element tego ciągu oraz liczbę jego elementów. Możesz założyć, że jest jeden taki ciąg.

Dla przykładowych danych:

23156
1231
1345
1456
1456
897

najdłuższy niemalejący ciąg liczb rozpoczyna się liczbą 1231 i składa się z 4 elementów.

62.3.

Porównaj wartości liczb zapisanych w wierszach o tych samych numerach w plikach `liczby1.txt` i `liczby2.txt`. Podaj liczbę wierszy, w których:

a) liczby mają w obu plikach taką samą wartość;

b) wartość liczby z pliku `liczby1.txt` jest większa od wartości liczby z pliku `liczby2.txt`.

Dla przykładowych danych:

```
liczby1.txt    liczby2.txt
11456          1302
22666          9654
546            499
```

odp. a) 1 wiersz, bo tylko w drugim wierszu liczby mają taką samą wartość: $22666_8 = 9654_{10}$

odp. b) 1 wiersz, bo tylko w pierwszym wierszu wartość liczby w pierwszym pliku jest większa niż odpowiadająca jej wartość w drugim pliku: $11456_8 > 1302_{10}$.

62.4.

Podaj, ile razy w zapisie dziesiętnym wszystkich liczb z pliku `liczby2.txt` występuje cyfra 6 oraz ile razy wystąpiłaby ta cyfra, gdyby te same liczby były zapisane w systemie ósemkowym.

Zadanie 63.

Wiązka zadań *Ciągi zerojedynkowe*

W pliku `ciagi.txt` w oddzielnych wierszach znajduje się **1000 różnych ciągów zerojedynkowych**, każdy o długości od 2 do 18. **Napisz program(-y)**, który pozwoli rozwiązać poniższe zadania. Następnie je rozwiąż, a odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym `wyniki_ciagi.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

63.1.

Ciągiem dwucyklicznym będziemy nazywać taki ciąg zerojedynkowy w o długości parzystej, który składa się z dwóch fragmentów w_1 oraz w_2, $w = w_1 w_2$, takich że $w_1 = w_2$. Podaj wszystkie ciągi dwucykliczne zapisane w pliku `ciagi.txt`.

Przykład

Dla zestawu ciągów:

```
100101010100110010101001
11001101001
1001000
11001100
101010011100
110011110011
```

3 podkreślone ciągi są dwucykliczne.

63.2.

Podaj liczbę ciągów z pliku `ciagi.txt`, w których nie występują obok siebie dwie jedynki.

Przykład

Dla zestawu ciągów:

```
101010101001010010010101
11001101001
10001000
101010011100
000011
```

wynikiem jest liczba 2 (w dwóch podkreślonych ciągach dwie jedynki nie występują obok siebie).

63.3.

Liczbą półpierwszą nazywamy taką liczbę, która jest iloczynem dwóch liczb pierwszych. Podaj, ile ciągów z pliku `ciagi.txt` jest reprezentacją binarną liczb półpierwszych. Dodatkowo podaj największą i najmniejszą liczbę półpierwszą w zapisie dziesiętnym.

Przykład

Dla zestawu ciągów:

```
100010
1101001001
1100101
1111111111
10010110000010010010
```

podkreślone ciągi są zapisem binarnym liczb półpierwszych, ponieważ:

$(100010)_2 = 34 = 2 * 17$, więc jest liczbą półpierwszą;

$(1101001001)_2 = 841 = 29 * 29$, więc jest liczbą półpierwszą;

$(1100101)_2 = 101 = 101 * 1$;

$(1111111111)_2 = 1023 = 3 * 11 * 31$;

$(10010110000010010010)_2 = 614546 = 2 * 307273$, więc jest liczbą półpierwszą.

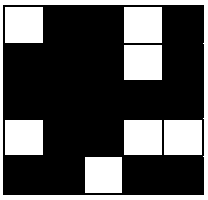
Zadanie 64.

Wiązka zadań *Obrazki*

Bit parzystości ciągu złożonego z zer i jedynek jest równy 0, gdy w ciągu tym występuje parzysta liczba jedynek, w przeciwnym razie bit parzystości jest równy 1.

Czarno-biały obrazek rozmiaru $n \times n$ składa się z n wierszy po n pikseli. Każdy wiersz pikseli reprezentujemy jako ciąg zer i jedynek, każdy biały piksel reprezentujemy przez 0, czarny — przez 1. Na końcu każdego wiersza dodany jest bit parzystości, podobnie pod ostatnim wierszem obrazka dołączony jest wiersz bitów parzystości każdej z n kolumn. **Bitów parzystości nie traktujemy jako części obrazka.**

Przykład: Poniżej podajemy obrazek rozmiaru 5×5 oraz jego reprezentację, wraz z odpowiednimi bitami parzystości (bity parzystości zostały podkreślone):



Obrazek

```
0 1 1 0 1 1
1 1 1 0 1 0
1 1 1 1 1 1
0 1 1 0 0 0
1 1 0 1 1 0
1 1 0 0 0
```

Reprezentacja

Plik `dane_obrazki.txt` składa się z opisu 200 czarno-białych obrazków o rozmiarze 20×20 pikseli. **Sąsiednie obrazki oddzielone są w pliku pustym wierszem.**

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w pliku `wyniki_obrazki.txt`. Odpowiedź do każdego zadania rozpocznij w nowym wierszu, poprzedzając ją numerem zadania.

64.1.

Obrazek nazywamy *rewersem*, jeśli liczba występujących w nim pikseli czarnych jest większa od liczby pikseli białych.

Przykład: W obrazku z powyższego przykładu występuje 18 pikseli czarnych i 7 pikseli białych. Zatem jest on rewersem.

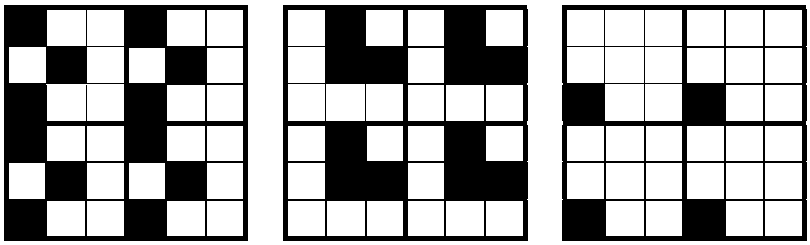
Podaj, ile jest w pliku obrazków, które są rewersami. Podaj też największą liczbę pikseli czarnych występujących w jednym obrazku.

64.2.

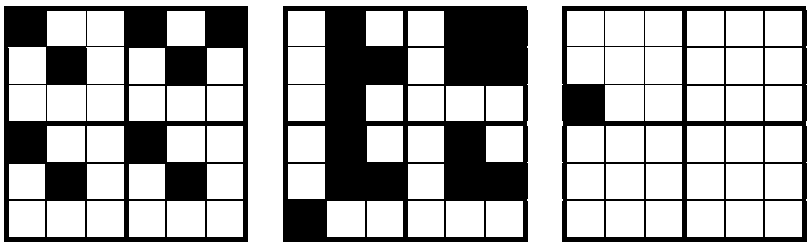
Obrazek rozmiaru $n \times n$ będziemy nazywać *rekurencyjnym,* jeśli n jest parzyste oraz obrazek składa się z 4 kopii tego samego obrazka rozmiaru $\frac{n}{2} \times \frac{n}{2}$.

Przykład

Poniżej podajemy 3 obrazki rozmiaru 6×6, które są rekurencyjne.



Natomiast poniższe obrazki nie są rekurencyjne:



Podaj liczbę obrazków rekurencyjnych w pliku wejściowym. Ponadto podaj opis pierwszego obrazka rekurencyjnego występującego w pliku. W opisie obrazka pomiń bity parzystości (pamiętaj, że obrazek składa się z 20 wierszy po 20 pikseli, które reprezentujemy jako ciąg zer i jedynek).

64.3.

Obrazek nazywamy *poprawnym*, jeśli wszystkie bity parzystości są w nim poprawne (zarówno w wierszach, jak i kolumnach). Obrazek nazywamy *naprawialnym*, jeśli nie jest poprawny, a jednocześnie co najwyżej jeden bit parzystości wiersza i co najwyżej jeden bit parzystości kolumny jest w nim niepoprawny.

Natomiast *nienaprawialnym* nazywamy obrazek, który nie jest poprawny i nie jest naprawialny.

Przykład

Poniżej podajemy przykłady obrazków poprawnych, naprawialnych i nienaprawialnych rozmiaru 5×5. Niepoprawne bity parzystości w obrazkach zostały wyróżnione podkreśleniem.

011000	011011	011011	011011	011011
111010	11101**1**	11101**1**	11101**1**	11101**1**
111111	111111	111111	111111	111111
011000	011000	011000	01100**1**	01100**1**
110011	110110	110110	110110	110110
11011	11**1**00	11000	11000	**0**1000
poprawny	naprawialny	naprawialny	nienaprawialny	nienaprawialny

Podaj liczbę obrazków poprawnych, liczbę obrazków naprawialnych oraz liczbę obrazków nienaprawialnych. Ponadto podaj największą liczbę błędnych bitów parzystości występujących w jednym obrazku.

64.4.

W obrazku naprawialnym wystarczy zmienić jedną wartość, aby uzyskać obrazek poprawny. Dokładniej, jeśli niepoprawne są bity parzystości *i*-tego wiersza i *j*-tej kolumny, wystarczy zmienić *j*-ty piksel w *i*-tym wierszu. Jeśli niepoprawny jest dokładnie jeden bit parzystości (wiersza **albo** kolumny), wystarczy zmienić ten bit parzystości.

Przykład

Rozważmy następujące dwa obrazki naprawialne rozmiaru 5×5 (niepoprawne bity parzystości w obrazkach zostały podkreślone).

Obraz 1:

011011
11**1**01**1**
111111
011000
110110
11**1**00

Obraz 2:

011011
11101**1**
111111
011000
110110
11000

Zmieniając trzecią jedynkę w drugim wierszu pierwszego obrazka, uzyskamy obrazek poprawny (niepoprawne były bity parzystości trzeciej kolumny i drugiego wiersza). Zmieniając niepoprawny bit parzystości w drugim wierszu drugiego obrazka z 1 na 0, również uzyskamy obrazek poprawny.

Podaj numery obrazków naprawialnych, przyjmując, że numery kolejnych obrazków w pliku to 1, 2, 3 itd. Przy numerze każdego obrazka naprawialnego podaj numer wiersza i kolumny wartości, którą wystarczy zmienić, aby uzyskać obrazek poprawny.

Zadanie 65.

Wiązka zadań *Ułamki*

W pliku `dane_ulamki.txt` znajduje się 1000 par liczb naturalnych dodatnich, mniejszych niż 12 000. Każda para liczb jest zapisana w osobnym wierszu, liczby w wierszu rozdzielone są pojedynczym znakiem odstępu. Parę liczb zapisanych w tym samym wierszu interpretujemy jako ułamek, którego licznikiem jest pierwsza liczba, a mianownikiem — druga liczba.

Przykład

Plik o zawartości

```
3 6
2 3
5 3
2 4
15 5
```

odpowiada ułamkom $\frac{3}{6},\frac{2}{3},\frac{5}{3},\frac{2}{4},\frac{15}{5}$.

Postacią nieskracalną ułamka $\frac{a}{b}$ nazywamy taki ułamek $\frac{x}{y}$, że $\frac{a}{b}=\frac{x}{y}$ oraz x i y są względnie pierwsze (czyli x i y nie mają wspólnego dzielnika większego od 1).

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w pliku `wyniki_ulamki.txt`. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją numerem zadania.

65.1.

Podaj ułamek o minimalnej wartości. Jeśli w pliku występuje więcej niż jeden taki ułamek, to podaj ten spośród nich, który ma najmniejszy mianownik. Twoja odpowiedź powinna zawierać parę liczb oznaczającą licznik i mianownik ułamka.

Przykład

Dla podanego powyżej pliku, opisującego ułamki $\frac{3}{6},\frac{2}{3},\frac{5}{3},\frac{2}{4},\frac{15}{5}$, minimalną wartość mają ułamki $\frac{3}{6},\frac{2}{4}$. Ponieważ $\frac{2}{4}$ ma mniejszy mianownik niż $\frac{3}{6}$, więc odpowiedzią jest para liczb: 2 i 4.

65.2.

Podaj liczbę zapisanych w pliku ułamków, które zostały podane w postaci nieskracalnej.

Przykład

Dla podanego powyżej pliku, opisującego ułamki $\frac{3}{6},\frac{2}{3},\frac{5}{3},\frac{2}{4},\frac{15}{5}$, w postaci nieskracalnej zapisane zostały $\frac{2}{3},\frac{5}{3}$. Natomiast $\frac{3}{6}$ i $\frac{2}{4}$ nie są ułamkami w postaci nieskracalnej (ich liczniki i mianowniki dzielą się odpowiednio przez 3 i 2), podobnie $\frac{15}{5}$ (jego licznik i mianownik dzielą się przez 5). Zatem odpowiedzią jest liczba 2.

65.3.

Zapis danych w postaci nieskracalnej uzyskamy, zamieniając każdy ułamek na jego postać **nieskracalną**. Podaj sumę liczników wszystkich podanych w pliku ułamków, jaką otrzymalibyśmy po sprowadzeniu ułamków do nieskracalnej postaci.

Przykład

Dla podanego powyżej pliku, opisującego ułamki $\frac{3}{6},\frac{2}{3},\frac{5}{3},\frac{2}{4},\frac{15}{5}$, nieskracalne postacie kolejnych ułamków to: $\frac{1}{2},\frac{2}{3},\frac{5}{3},\frac{1}{2},\frac{3}{1}$. Suma liczników tych ułamków to 1+2+5+1+3=12. Zatem odpowiedzią jest 12.

65.4.

Ułamki w pliku zostały tak dobrane, że każdy mianownik jest dzielnikiem liczby $b=2^2\cdot3^2\cdot5^2\cdot7^2\cdot13$, a wartość każdego ułamka jest nie większa niż 3. Oznacza to, że sumę wszystkich ułamków można przedstawić jako ułamek $\frac{a}{b}$, którego mianownikiem jest $b=2^2\cdot3^2\cdot5^2\cdot7^2\cdot13$. Wyznacz sumę ułamków ze wszystkich wierszy i podaj licznik takiego ułamka, że suma ułamków jest równa $\frac{a}{b}$.

Przykład

Dla podanego powyżej pliku, opisującego ułamki 1/2, 2/3, 5/3, 2/4, 15/5, suma ułamków to:

$$\frac{1}{2}+\frac{2}{3}+\frac{5}{3}+\frac{2}{4}+\frac{15}{5}=\frac{\frac{b}{2}+\frac{2b}{3}+\frac{5b}{3}+\frac{2b}{4}+\frac{15b}{5}}{2^2\cdot3^2\cdot5^2\cdot7^2\cdot13}=\frac{3630900}{2^2\cdot3^2\cdot5^2\cdot7^2\cdot13},$$

gdzie $b=2^2\cdot3^2\cdot5^2\cdot7^2\cdot13$. Poprawna odpowiedź wynosi więc 3630900.

Zadanie 66.

Wiązka zadań *Trójki liczb*

W pliku `trojki.txt` w oddzielnych wierszach znajduje się 1000 trójek liczb naturalnych z przedziału od 1 do 550000000. W każdym wierszu są umieszczone trzy liczby rozdzielone pojedynczymi odstępami.

Przykład

3 4 5

12 5 13

12 491 17

11 13 143

15 28 91

Napisz program(-y), który da odpowiedzi do poniższych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym `wyniki_trojki.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

66.1.

Wypisz wszystkie trójki liczb z pliku `trojki.txt`, w których suma cyfr dwóch pierwszych liczb jest równa ostatniej (trzeciej) liczbie.

Przykład

12 491 17

1+2+4+9+1=17

66.2.

Wypisz wszystkie wiersze z pliku `trojki.txt` zawierające takie trzy liczby *a*, *b*, *c*, w których *a* i *b* są liczbami pierwszymi oraz $c = a \cdot b$.

Przykład

11 13 143

11 i 13 są liczbami pierwszymi i $11 \cdot 13 = 143$

66.3.

Wypisz z pliku `trojki.txt` wszystkie pary sąsiadujących ze sobą wierszy, takie że liczby w tych wierszach są długościami boków trójkątów prostokątnych.

Przykład

3 4 5 $3^2+4^2=5^2$

12 5 13 $5^2+12^2=13^2$

66.4.

Podaj, ile jest w pliku `trojki.txt` wierszy, w których znajdują się liczby reprezentujące długości boków trójkąta. Ciąg wierszy nazywamy **trójkątnym**, jeśli liczby w każdym wierszu reprezentują długości boków trójkąta. Podaj długość najdłuższego ciągu trójkątnego w pliku.

Zadanie 67.

Wiązka zadań *Binarny fraktal Fibonacciego*

Ciąg Fibonacciego to ciąg liczb naturalnych określony rekurencyjnie w sposób następujący:

$F_1 = 1, F_2 = 1$, a każdy następny element ciągu jest sumą dwóch poprzednich, czyli:

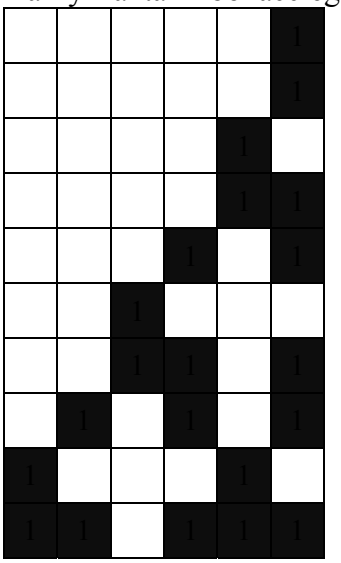
$$F_n = \begin{cases} 1 & dla\ n = 1 \\ 1 & dla\ n = 2 \\ F_{n-1} + F_{n-2} & dla\ n > 2 \end{cases}$$

Binarny fraktal Fibonacciego to dwuwymiarowa tablica zawierająca w kolejnych wierszach binarne zapisy kolejnych liczb Fibonacciego, gdzie każde zero w zapisie zastąpiono białym kwadratem, a każdą jedynkę czarnym kwadratem (p. rysunek). Wszystkie binarne zapisy powinny składać się z jednakowej liczby cyfr, czyli do zapisów krótszych niż najdłuższy należy dodać zera wiodące.

Przykład binarnego fraktala dla pierwszych 10 liczb Fibonacciego:

n	F_n	zapis binarny F_n
1	1	000001
2	1	000001
3	2	000010
4	3	000011
5	5	000101
6	8	001000
7	13	001101
8	21	010101
9	34	100010
10	55	110111

Binarny fraktal Fibonacciego



Napisz program komputerowy, za pomocą którego uzyskasz odpowiedzi do poniższych zadań. Rysunek fraktala (zadanie nr 3) wykonaj, wykorzystując dostępne narzędzia informatyczne. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki.txt`, natomiast rysunek fraktala w pliku `fraktal.xxx`, gdzie `xxx` oznacza rozszerzenie pliku, w którym zapisany jest obraz fraktala.

67.1.

Podaj wartości F_{10}, F_{20}, F_{30}, F_{40}. Zapisz każdą z liczb w osobnym wierszu.

67.2.

Znajdź wszystkie liczby pierwsze wśród liczb F_1, F_2, … , F_{40}. Zapisz każdą z liczb w osobnym wierszu.

67.3.

Dla pierwszych 40 liczb Fibonacciego utwórz binarny fraktal Fibonacciego:

- Wypisz reprezentację binarną wszystkich liczb Fibonacciego od F_1 do F_{40}.
- Wyrównaj długości reprezentacji binarnych wszystkich liczb Fibonacciego od F_1 do F_{40} i na ich podstawie sporządź obraz binarnego fraktala Fibonacciego.

67.4.

Podaj w zapisie binarnym wyrazy ciągu Fibonacciego z zakresu od F_1 do F_{40}, które w tym zapisie mają dokładnie 6 jedynek.

Zadanie 68.

Wiązka zadań *Napisy — anagramy*

Dwa napisy *a* i *b* są swoimi **anagramami**, jeżeli napis *a* (napis *b*) można zapisać za pomocą liter występujących w napisie *b* (napisie *a*), wykorzystując **wszystkie** jego litery.

W pliku `dane_napisy.txt` znajduje się 1000 par napisów, z których każdy jest długości od 2 do 20 znaków, składających się z wielkich liter: A, B, C, D, E, F, G, H, I, J. Każda para napisów jest zapisana w osobnym wierszu, a napisy oddzielone są pojedynczym znakiem odstępu.

Przykład

```
AIHAHGHBEAFJAJDI HGIHFEHHJGBCBGD
FBJHCFFGADD EHADJAJBJBEGD
JHGHADJ AGFEHHEHIAEJFC
EJJHFHIGCEBDAIB DCAFFDICGBEAHAEJ
FBAEEGICHFFACICIGB EEHAHHCABHDHGDFFEGD
```

Napisz program(-y), który poda odpowiedzi dla następujących zadań. Odpowiedzi zapisz w pliku `wyniki_anagramy.txt`. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją identyfikatorem zadania.

68.1.

Napis nazywamy **jednolitym**, jeżeli wszystkie jego litery są takie same. Przykładem takiego napisu jest *AAAA*. Podaj liczbę wierszy zawierających parę napisów jednolitych, które są wzajemnie swoimi anagramami.

Przykład

Dla pliku zawierającego następujące dane:

```
AAAA AAAA
AHHAH AHHAH
AAAA AAAAAAA
BBBBBBB BABBAB
CCCCC CCCCC
```

wynikiem jest liczba 2 (pierwszy i ostatni wiersz). Zwróć uwagę, że napisy w trzecim wierszu są napisami jednolitymi, ale nie są wzajemnie swoimi anagramami.

68.2.

Podaj liczbę wierszy, które zawierają napisy będące wzajemnie swoimi **anagramami**.

Przykład

Dla pliku zawierającego następujące dane:

```
BBBAAB BBBABA
AAAA AAAAA
AHHAH AHHAH
BBABBABB BBBABB
BABABB CACACC
```

wnikiem jest liczba 2 (pierwszy i trzeci wiersz).

68.3.

Podaj największą liczbę *k* taką, że w pliku znajduje się *k* napisów, z których każde dwa są wzajemnie swoimi anagramami.

Przykład

Dla pliku zawierającego następujące dane:

BABABB BBBABA
AAAA AAAA
AHHAH AHHAH
BBABBABB BABBAB
BBAABB CCCCC

wnikiem jest liczba 4 (*BABABB BBBABA BABBAB BBAABB*).

Zadanie 69.

Wiązka zadań *Geny*

Informację genetyczną (genotyp) każdego osobnika z galaktyki Madgen opisuje słowo (napis), w którym występują litery ze zbioru {A, B, C, D, E}. Obowiązują przy tym następujące zasady:

1. Organizmy żyjące na Madgen tworzą gatunki $g_1, g_2, g_3, \ldots$, gdzie g_i to zbiór osobników o długości genotypu równej *i*.
2. W skład genotypu mogą wchodzić geny. Pierwszy gen rozpoczyna się pierwszą występującą w genotypie sekwencją AA, a kończy się najbliższą napotkaną po niej sekwencją BB. Każdy kolejny gen rozpoczyna się pierwszą sekwencją AA, występującą za końcem poprzedniego genu, i analogicznie kończy się najbliższą napotkaną sekwencją BB.
3. Geny nazywamy częścią kodującą genotypu, pozostałe fragmenty tworzą część niekodującą.

Przykład 1.

Genotyp AACDBABBBCDAABCBBAAE

zawiera geny AACDBABB oraz AABCBB. Zwróćmy uwagę, że:

- ciąg AA występujący za genem AABCBB nie jest początkiem genu, ponieważ nie występuje za nim ciąg BB kończący gen;
- część kodująca genotypu AACDBABBBCD***AABCBB***AAE jest równa AACDBABB***AABCBB***.

Przykład 2.

Genotyp AADBAADDDDEEEBBEE

zawiera gen **AA**DB**AA**DDDDEEEBB. Zwróćmy uwagę, że:

- pierwsze pojawienie się ciągu AA determinuje początek genu, dlatego w powyższym genotypie występuje gen AADBAADDDDEEEBB, a nie gen AADDDDEEEBB.

Plik `dane_gen.txt` zawiera genotypy 1000 osobników z galaktyki Madgen. Każdy wiersz pliku zawiera genotyp jednego osobnika o długości nie większej niż 500 znaków.

Przykład

```
ABAEACBAADAACAABBABCDA
ABAEACBADEADACACABBABCDA
```

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w pliku `wyniki_gen.txt`. Odpowiedź do każdego zadania rozpocznij w nowym wierszu, poprzedzając ją numerem zadania.

69.1.

Podaj liczbę wszystkich gatunków, których genotypy zapisane są w pliku `dane_gen.txt`. Podaj największą liczbę osobników reprezentujących ten sam gatunek.

69.2.

Występowanie w jakimkolwiek **genie** ciągu **`BCDDC`** oznacza mutację powodującą małą odporność osobnika na zmęczenie. Podaj, ile osobników spośród tych, których genotypy zapisane są w pliku, ma tę mutacje.

Przykład

Osobnik o genie AAC**`BCDDC`**BBACDE ma mutację BCDDC (ciąg BCDDC występuje w obrębie podkreślonego genu), natomiast osobnik o genie C**`BCDDC`**BBAACDEBB nie ma tej mutacji, gdyż występujący ciąg BCDDC nie jest ulokowany w żadnym genie.

69.3.

Wyznacz i podaj największą liczbę genów występujących u jednego osobnika. Podaj też największą długość genu zapisanego w całym pliku.

Przykład

Rozważmy plik składający się z genotypów:

```
EAABCDBBDCBAAE
EAABCDBBDCBAAEBCDBBEE
EAABCDBBECAAB
```

Pierwszy osobnik ma jeden gen (AABCDBB), drugi ma dwa geny (AABCDBB i AAEBCDBB), a trzeci osobnik ma jeden gen (AABCDBB). Zatem największa liczba genów u jednego osobnika wynosi 2, a największa długość genu to 8 (gen AAEBCDBB ma tę długość).

69.4.

Genotyp odczytywany z materiału biologicznego może być odkodowany w kierunku od strony lewej do prawej lub odwrotnie: od strony prawej do lewej. Genotyp nazywać będziemy *odpornym*, jeśli czytany od strony lewej do prawej oraz od strony prawej do lewej ma dokładnie taką samą część kodującą. Natomiast genotyp *silnie odporny* to taki, który czytany od strony lewej do prawej oraz od strony prawej do lewej daje dokładnie ten sam napis. (Inaczej mówiąc, genotyp jest silnie odporny, gdy jest palindromem).

Przykład

Rozważmy genotypy:

```
EAABCDBBDCBAAE
EAABCDBBDCBAAEBCDEE
EAABCDBBECAAB
```

Genotyp `EAABCDBBDCBAAE` jest silnie odporny (jest palindromem). Genotyp `EAABCDBBDCBAAEBCDEE` nie jest silnie odporny (nie jest palindromem), ale jest odporny, gdyż czytany od strony lewej do prawej, jak i od strony prawej do lewej ma taką samą część kodującą: `AABCDBB`. Natomiast genotyp `EAABCDBBECAAB` nie jest silnie odporny (nie jest palindromem), nie jest też odporny, gdyż czytany od strony lewej do prawej daje część kodującą `AABCDBB`, a czytany od strony prawej do lewej ma część kodującą równą `AACEBB`.

Wyznacz liczbę genotypów odpornych oraz liczbę genotypów silnie odpornych.

Zadanie 70.

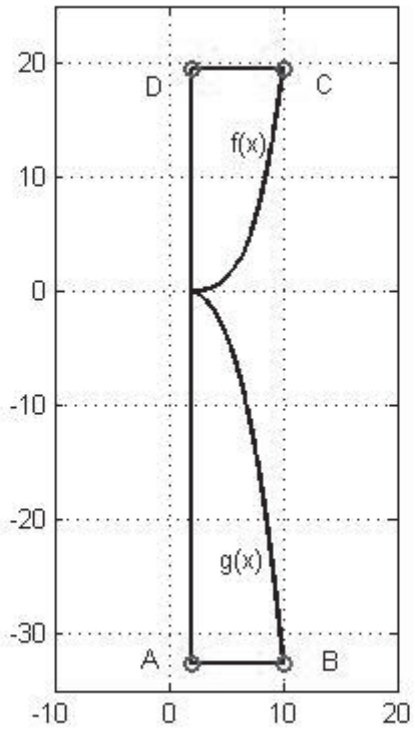
Wiązka zadań *Zasłona*

Pani Binarna dostała zlecenie na uszycie zasłony. Na rysunku poniżej przedstawiono zasłonę, która jest ograniczona:

- od góry prostą $y = 19\frac{61}{125}$,
- od dołu prostą $y = -32\frac{2}{3}$,
- z lewej strony prostą $x = 2$,
- z prawej strony dwoma krzywymi: $f(x) = \frac{x^4}{500} - \frac{x^2}{200} - \frac{3}{250}$ oraz $g(x) = -\frac{x^3}{30} + \frac{x}{20} + \frac{1}{6}$.

Uwaga: Zauważ, że $f(10) = 19\frac{61}{125}$, zaś $g(10) = -32\frac{2}{3}$.

Rysunek pomocniczy:



Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe zadania. Odpowiedzi do nich umieść w pliku `zadanie_zaslona.txt`. Każda odpowiedź powinna być poprzedzona numerem je oznaczającym.

70.1.

Pani Binarna zakupiła tyle materiału, ile wynosi pole prostokąta ABCD, w którym mieści się zasłona. Oblicz, jaka będzie powierzchnia materiału pozostałego po wykrojeniu zasłony. Wynik podaj z dokładnością do 1/1000.

70.2.

Pani Binarna zamierza obszyć taśmą zasłonę ze wszystkich czterech stron, w tym celu chce wyznaczyć obwód zasłony. Część obwodu ograniczoną wykresem funkcji $f(x)$ szacujemy w następujący sposób: Odcinek [2,10] dzielimy na 1000 równych części, których prawe końce oznaczamy przez $x_1,\ldots,x_{1000}$. Długość krzywej odpowiadającej wykresowi $f(x)$ na przedziale [2,10] przybliżamy długością łamanej łączącej punkty $(2, f(2))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$ itd. aż do $(x_{1000}, f(x_{1000}))$. Analogicznie wyznaczamy część obwodu ograniczoną przez $g(x)$.

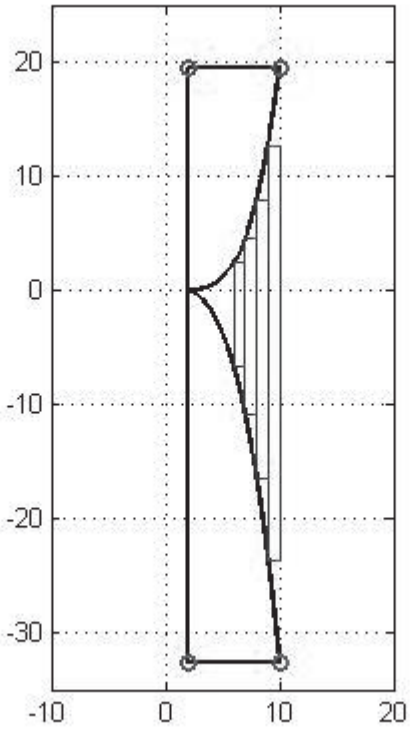
Stosując powyższą metodę wyznaczania obwodu, oblicz długość taśmy, jaką musi zakupić pani Binarna, zakładając, że w sprzedaży jest tylko taśma o długościach będących wielokrotnością jednego metra.

70.3.

Pani Binarna postanowiła wykorzystać pozostały fragment materiału i wyciąć z niego pasy o szerokości 0,25 m i o bokach równoległych do osi układu współrzędnych. Podaj sumę długości pasów, które można wyciąć z pozostałego fragmentu materiału. Załóż, że długość każ-

dego wyciętego pasa jest liczbą całkowitą oraz że pani Binarna zaczyna wycinać pasy od prawej strony materiału.

Rysunek pomocniczy:



Zadanie 71.

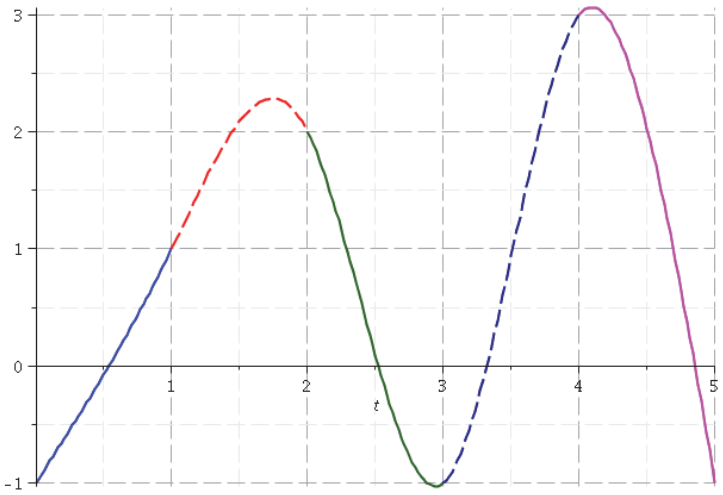
Wiązka zadań *Funkcja*

Wykres funkcji *f* złożony jest z pięciu fragmentów:

$$f(x) = \begin{cases} f_1(x), & x \in [0,1), \\ f_2(x), & x \in [1,2), \\ f_3(x), & x \in [2,3), \\ f_4(x), & x \in [3,4), \\ f_5(x), & x \in [4,5), \end{cases}$$

gdzie każda z funkcji $f_i(x)$ jest wielomianem stopnia trzeciego. W pliku `funkcja.txt` zapisane są współczynniki postaci ogólnej wielomianów $f_i(x)(i = 1,2,\dots,5)$; w *i*-tym wierszu pliku zapisane są cztery liczby rzeczywiste: a_0, a_1, a_2, a_3 (oddzielone pojedynczym odstępem), dla których $f_i(x) = a_0 + a_1x + a_2x^2 + a_3x^3$.

Poniższy rysunek przedstawia wykres funkcji f.



Napisz program, który da odpowiedzi do poniższych zadań. Zapisz je w pliku tekstowym `wyniki_funkcja.txt`. Wyniki do każdego zadania poprzedź numerem je oznaczającym. Wszystkie wyniki należy wypisać, stosując zaokrąglenie do podanej liczby cyfr po przecinku.

71.1.

Podaj wartość $f(1.5)$ z dokładnością do 5 cyfr po przecinku.

71.2.

Znajdź wartość $x \in [0,5)$, dla której wartość $f(x)$ jest największa.

Jako wynik podaj wartość x z dokładnością do trzech, a wartość $f(x)$ z dokładnością do 5 cyfr po przecinku.

71.3.

Znajdź wszystkie miejsca zerowe funkcji f w przedziale $[0,5)$. Odpowiedzi podaj z dokładnością do 5 cyfr po przecinku.

Zadanie 72.
Wiązka zadań *Podobne napisy*

W pliku `napisy.txt` znajduje się 200 wierszy, z których każdy zawiera dwa napisy o długości od 1 do 50 znaków, oddzielone pojedynczym odstępem. Napisy składają się wyłącznie z małych liter alfabetu angielskiego.

Napisz program (lub kilka programów), który pozwoli rozwiązać poniższe zadania. Odpowiedzi zapisz w pliku `wyniki.txt`.

72.1.

Oblicz, w ilu wierszach jeden (którykolwiek) z napisów jest przynajmniej trzy razy dłuższy od drugiego. Jako odpowiedź wypisz liczbę takich wierszy oraz parę napisów z pierwszego z nich.

72.2.

Znajdź (i wypisz) wszystkie takie wiersze pliku, w których drugi napis da się otrzymać z pierwszego przez dopisanie na jego końcu pewnej dodatniej liczby liter (na przykład *kot*

i *kotara*). Dla każdego wiersza podaj oba znajdujące się w nim napisy, a osobno wypisz litery, które należy dopisać.

72.3.

Niektóre z podanych par napisów mają identyczne zakończenia (na przykład *komputer* i *krater*). Znajdź i wypisz największą możliwą długość takiego zakończenia, a także wszystkie pary napisów w wierszach, które osiągają tę maksymalną długość.

Zadanie 73.
Wiązka zadań *Statystyki tekstu*

W pliku `tekst.txt` dany jest tekst złożony ze słów pisanych wielkimi literami alfabetu angielskiego. Słów jest 1876, oddzielone są one pojedynczymi odstępami, a inne znaki poza literami i spacjami w tekście nie występują. Napisz program(-y), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku `wyniki.txt`.

73.1.

Oblicz, ile jest w tekście słów, w których występują dwie kolejne takie same litery.

73.2.

Sporządź statystykę częstotliwości występowania liter w tekście: dla każdej litery podaj liczbę jej wystąpień we wszystkich słowach tekstu oraz jej procentowy udział wśród wystąpień wszystkich liter w tekście(do statystyki nie wliczaj spacji). Odpowiedź zapisz w następującej postaci:

```
A: 632 (7.56%)
B: 196 (2.34%)
...
```

Wartości procentowe podaj w zaokrągleniu do dwóch miejsc po przecinku.

73.3.

Wśród słów w tekście policz długość najdłuższego podsłowa (fragmentu złożonego z kolejnych liter) złożonego z samych spółgłosek. Pamiętaj, że samogłoski to: A, E, I, O, U oraz Y, zaś pozostałe litery są spółgłoskami.

Podaj długość najdłuższego takiego podsłowa, liczbę słów, które zawierają podsłowo o takiej długości, oraz pierwsze z nich, które występuje w pliku `tekst.txt`.

Zadanie 74.
Wiązka zadań *Hasła*

W pliku `hasla.txt` danych jest 200 haseł użytkowników pewnego systemu. Każdy użytkownik posiada jedno hasło (każde zapisane jest w osobnym wierszu), które zawiera od 1 do 20 znaków alfanumerycznych, tzn. cyfr od 0 do 9 lub liter alfabetu łacińskiego (małych lub dużych). Polityka bezpieczeństwa systemu wymaga, aby hasła były odpowiednio skomplikowane i nie powtarzały się.

Poniżej podano pierwsze pięć haseł zapisanych w pliku `hasla.txt`:

```
ZXUhkPLcjKo
ikfLDegQXj
8Y7JGYXXR5
603624722555
50q4252ax5
```

Napisz program, który da odpowiedzi do poniższych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym `wyniki_hasla.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

74.1.

Podaj liczbę haseł złożonych jedynie ze znaków numerycznych, tzn. cyfr od 0 do 9.

74.2.

Wypisz hasła, które zostały użyte przez co najmniej dwóch różnych użytkowników, tzn. występujące w dwóch różnych wierszach. Hasła wypisz (bez powtórzeń) w kolejności leksykograficznej.

74.3.

Podaj liczbę użytkowników posiadających hasła, w których występuje fragment złożony z czterech kolejnych znaków ASCII (w dowolnej kolejności).

Przykłady haseł zawierających taki fragment to:

```
A5mnpoR89cd
A5876RRcg
As45FGHFEk90nba
```

75.4.

Podaj liczbę haseł, które spełniają jednocześnie poniższe warunki:

- hasło zawiera co najmniej jeden znak numeryczny, tzn. cyfrę od 0 do 9,
- hasło zawiera co najmniej jedną małą literę,
- hasło zawiera co najmniej jedną dużą literę.

Zadanie 75.

Wiązka zadań *Szyfr afiniczny*

Dany jest tekst złożony ze słów zbudowanych z małych liter alfabetu angielskiego. Metoda szyfrowania afinicznego — dla której *kluczem szyfrującym* są dwie liczby całkowite A i B — polega na wykonaniu kolejno następujących operacji:

- zastąpienia kolejnych liter alfabetu liczbami od 0 do 25: 'a' przez 0, 'b' przez 1, 'c' przez 2 itd. według przyporządkowania przedstawionego w poniższej tabeli:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- pomnożenia liczby odpowiadającej każdej literze przez A i dodania otrzymanego wyniku do B,

- zamiany otrzymanych liczb z powrotem na litery; jeśli liczba jest większa niż 25, bierze się jej resztę z dzielenia przez 26.

Parametry klucza, czyli liczby A i B, powinny być liczbami całkowitymi z przedziału [0, 25].

Dla przykładu, jeśli kluczem szyfrującym jest (3, 7), czyli A = 3, zaś B = 7, to litera 'n' jest najpierw zastępowana liczbą 13. Po pomnożeniu jej przez A i dodaniu B otrzymujemy wynik równy 46. W następnym kroku otrzymujemy literę o numerze 46 – 26 = 20, czyli 'u'.

Okazuje się, że do odszyfrowania szyfru afinicznego można zastosować tę samą metodę, być może z innym kluczem. Na przykład, jeśli napis zaszyfrujemy kluczem (3, 7), to aby go odszyfrować, stosujemy ten sam algorytm z kluczem (9, 15). Dla przykładu, deszyfrując literę 'u' z kluczem (9, 15), otrzymamy liczbę 20 * 9 + 15 = 195, czyli literę 'n', jako że 195 mod 26 = 13. Klucz (9,15) jest wówczas *kluczem deszyfrującym* dla klucza (3,7).

Napisz program(y), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku `wyniki.txt`.

75.1.

W pliku `tekst.txt` dany jest, w pojedynczym wierszu, tekst złożony z dokładnie 805 słów zapisanych małymi literami alfabetu angielskiego, oddzielonych znakami odstępu. Żadne słowo nie jest dłuższe niż 15 znaków.

Znajdź i wypisz te słowa, których zarówno pierwszą, jak i ostatnia literą jest 'd'.

75.2.

Zaszyfruj szyfrem afinicznym o kluczu (5, 2) te słowa z pliku `tekst.txt`, które składają się z co najmniej 10 liter. Wypisz je w postaci zaszyfrowanej, po jednym w wierszu.

75.3.

Plik `probka.txt` składa się z 5 wierszy, każdego zawierającego dwa napisy. Pierwszy z nich to pewne słowo zapisane tekstem jawnym, drugi zaś to to samo słowo zaszyfrowane za pomocą szyfru afinicznego (każde słowo innym kluczem).

Dla każdego z tych słów znajdź i wypisz klucz szyfrujący oraz klucz deszyfrujący.

Zadanie 76.

Wiązka zadań *Szyfr*

Rozważamy szyfrowanie przestawieniowe, w którym kluczem jest *n-elementowa* tablica zawierająca różne liczby całkowite z przedziału [1, *n*]. Na przykład kluczem *5*-elementowym może być tablica [3, 2, 5, 4, 1].

Szyfrowanie napisu *A* (o długości co najmniej *n*) kluczem *n*-elementowym *P*[1..*n*] odbywa się w następujący sposób:

- pierwsza litera słowa *A* zamieniana jest miejscami z literą na pozycji *P*[1],
- następnie druga litera słowa *A* zamieniana jest z literą na pozycji *P*[2]
- itd.

Uzyskane na końcu słowo jest szyfrem napisu *A* z kluczem *P*.

Jeśli napis *A* ma więcej niż *n* liter, to po *n*-tym kroku powyższego algorytmu kolejną literę zamieniamy znów z literą na pozycji *P*[1] itd. Oznacza to, że w *i*-tym kroku zamieniamy litery na pozycjach *i* oraz *P*[1+(*i*-1) mod *n*].

Przykład

Poniższa tabelka ilustruje szyfrowanie słowa „INFORMATYKA” kluczem P równym [3, 2, 5, 4, 1]:

i	1	2	3	4	5	6	7	8	9	10	11
P[1+(i-1) mod n]	3	2	5	4	1	3	2	5	4	1	3
Słowo	I	N	F	O	R	M	A	T	Y	K	A
Krok 1	**F**	N	**I**	O	R	M	A	T	Y	K	A
Krok 2	F	**N**	I	O	R	M	A	T	Y	K	A
Krok 3	F	N	**R**	O	**I**	M	A	T	Y	K	A
Krok 4	F	N	R	**O**	I	M	A	T	Y	K	A
Krok 5	**I**	N	R	O	**F**	M	A	T	Y	K	A
Krok 6	I	N	**M**	O	F	**R**	A	T	Y	K	A
Krok 7	I	**A**	M	O	F	R	**N**	T	Y	K	A
Krok 8	I	A	M	O	**T**	R	N	**F**	Y	K	A
Krok 9	I	A	M	**Y**	T	R	N	F	**O**	K	A
Krok 10	**K**	A	M	Y	T	R	N	F	O	**I**	A
Krok 11	K	A	**A**	Y	T	R	N	F	O	I	**M**

Napis „KAAYTRNFOIM” jest zatem szyfrem napisu „informatyka” z kluczem [3, 2, 5, 4, 1].
Napisz program(-y), który da odpowiedzi do poniższych zadań.

76.1.

W pliku `szyfr1.txt` dane są:

- w wierszach o numerach od 1 do 6 — napisy złożone z 50 liter alfabetu łacińskiego;
- w wierszu nr 7 — klucz 50-elementowy; liczby oddzielone są pojedynczym odstępem.

Zaszyfruj wszystkie sześć napisów zgodnie z opisaną metodą. Wynik, czyli zaszyfrowane napisy, zapisz w osobnych wierszach w pliku `wyniki_szyfr1.txt`.

76.2.

W pliku `szyfr2.txt` dane są:

- w pierwszym wierszu — napis złożony z 50 liter alfabetu łacińskiego;
- w drugim wierszu — klucz 15-elementowy; liczby oddzielone są pojedynczym odstępem.

Zaszyfruj dany napis zgodnie z opisaną metodą. Wynik, czyli zaszyfrowany napis, zapisz w pliku `wyniki_szyfr2.txt`.

76.3.

W pliku `szyfr3.txt` dany jest napis złożony z 50 liter alfabetu łacińskiego. Napis ten powstał po zaszyfrowaniu pewnego napisu *A* kluczem [6, 2, 4, 1, 5, 3].

Podaj napis *A*. Wynik zapisz w pliku `wyniki_szyfr3.txt`.

Zadanie 77.

Wiązka zadań *Szyfr Vigenère'a*

W zadaniu rozważamy teksty zbudowane tylko z wielkich liter alfabetu angielskiego, znaków odstępu i znaków przestankowych (przecinek, kropka). Oto litery alfabetu i numery ich pozycji w alfabecie:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Szyfrowanie Vigenère'a polega na zastąpieniu każdej litery tekstu źródłowego literą odległą od niej cyklicznie w alfabecie o *k* pozycji.

Wartość *k* nie jest z góry ustalona dla całego tekstu źródłowego, lecz dla każdej litery w tekście jest określana osobno, w oparciu o słowo przyjęte jako klucz szyfrowania.

Przystępując do szyfrowania, należy przyporządkować kolejnym literom tekstu źródłowego kolejne litery klucza, chodząc po nim cyklicznie, jeśli jest krótszy od szyfrowanego tekstu. Znaki inne niż litery nie są szyfrowane, pomijamy je podczas przypisywania liter klucza. P**ozycja litery klucza** w alfabecie jest tą wartością ***k***, o jaką należy wykonać przesunięcie względem litery tekstu źródłowego w celu znalezienia odpowiadającej jej litery szyfru.

Przykład

tekst źródłowy: `"JEST OK"`, klucz: `"EWA"`

tekst źródłowy	J	E	S	T	spacja	O	K
Klucz	E	W	A	E		W	A
pozycja litery klucza	4	22	0	4		22	0
Szyfr	J→4 = **N**	E→22=**A**	S→0 = **S**	T→4 = **X**	spacja	O→22=**K**	K→0 = **K**

wynik szyfrowania: `"NASX KK"`.

Napisz w wybranym języku programowania program, który wyznaczy rozwiązania zadań podanych niżej. Wszystkie wyniki zapisz w pliku tekstowym `Vigenere_wyniki.txt`, wyraźnie oddzielając odpowiedzi do poszczególnych zadań. Do oceny oddaj ten plik oraz plik (pliki) zawierający reprezentację komputerową rozwiązania.

77.1.

W pliku `dokad.txt` znajduje się jeden wiersz z tekstem. Długość tekstu nie przekracza 1024 znaków. Należy zaszyfrować ten tekst metodą Vigenère'a, używając jako klucza słowa: "LUBIMYCZYTAC".

a) Podaj liczbę powtórzeń klucza niezbędną do zaszyfrowania całego tekstu źródłowego (uwzględniając w nich ostatnie rozpoczęte powtórzenie).

b) Podaj zaszyfrowany tekst i zapisz go w pliku z odpowiedziami.

77.2.

W pliku `szyfr.txt` zapisano dwa wiersze. W pierwszym wierszu znajduje się tekst zaszyfrowany metodą Vigenère'a. W drugim wierszu znajduje się klucz użyty do tego szyfrowania.

Szyfr zawiera wiele słów. Jego łączna długość nie przekracza 1024 znaków. Szyfrowaniu podlegały tylko wielkie litery tekstu, zaś odstępy i znaki przestankowe pozostały bez zmiany.

Odszyfruj tekst i umieść jego postać źródłową w pliku z odpowiedziami.

77.3.

a) Podaj liczby wystąpień poszczególnych liter A, B, ..., Z w treści szyfru zawartego w pierwszym wierszu pliku `szyfr.txt`.

b) Chcąc złamać szyfr Vigenère, nie znając klucza, w pierwszym kroku należy oszacować długość klucza (rozumianą jako liczba znaków). Istnieje przybliżony wzór na szacunkową długość klucza *d* danego szyfru Vigenère'a dla tekstu nad alfabetem 26-literowym. Oszacowanie jest tym lepsze, im dłuższy jest szyfr.

$$d = \frac{0{,}0285}{\kappa_o - 0{,}0385}$$

gdzie κ_o to indeks koincydencji znaków obliczany następująco:

$$\kappa_o = \frac{l_A * (l_A - 1) + l_B * (l_B - 1) + \cdots + l_Z * (l_Z - 1)}{n * (n - 1)}$$

n — łączna liczba wystąpień **wszystkich liter** w tekście szyfru (nie liczymy odstępów i znaków przestankowych),

$l_A, l_B, \ldots, l_Z$ — liczby wystąpień **poszczególnych liter** `A`, `B`, …, `Z` w tekście szyfru.

Wykorzystując powyższe wzory, wyznacz szacunkową długość klucza dla szyfru danego w pierwszym wierszu pliku `szyfr.txt` i porównaj z dokładną długością klucza umieszczonego w drugim wierszu tego pliku. Wypisz obie wartości, wartość szacunkową zaokrąglij do 2 cyfr po przecinku.

Zadanie 78.

Wiązka zadań *Podpis elektroniczny*

Bajtek otrzymał od przyjaciela 11 jawnych wiadomości. Do każdej wiadomości przyjaciel dołączył podpis elektroniczny.

Podpis elektroniczny jest **zaszyfrowanym skrótem** wiadomości. Przyjaciel Bajtka utworzył skrót za pomocą funkcji *skrot*(), która przekształca dowolną wiadomość w 8-znakowy napis, a następnie zaszyfrował ten skrót algorytmem *A* o sobie tylko znanym kluczu prywatnym (*e*,*n*). Opis obu algorytmów podano poniżej.

Bajtek chciałby być pewien, że zachowano:

- integralność danych (treść nie została zmieniona w trakcie przesyłania),
- uwierzytelnienie nadawcy (nikt się pod nadawcę nie podszył).

W tym celu powinien sprawdzić każdą wiadomość następująco:

- zaprogramować funkcję *skrot*(*wiadomosc*) i za jej pomocą utworzyć skrót wiadomości,
- odszyfrować skrót z podpisu elektronicznego algorytmem *A* przy pomocy ogólnie znanego klucza publicznego (***d***,***n***) o wartościach (**3**,**200**),

- porównać oba skróty: jeśli są identyczne, znaczy to, że wiadomość jest wiarygodna.

Pomóż Bajtkowi sprawdzić, czy otrzymane wiadomości są wiarygodne.

W pliku `wiadomosci.txt` znajduje się 11 wiadomości, każda w osobnym wierszu. Liczba znaków każdej wiadomości nie przekracza 255. Wiadomości zawierają znaki pojedynczego odstępu, które są integralną częścią informacji.

W pliku `podpisy.txt` znajduje się 11 wierszy, każdy z nich zawiera 8 liczb całkowitych, stanowiących elementy podpisu elektronicznego jednej wiadomości. Liczby w wierszu oddzielone są pojedynczymi znakami odstępu. Kolejność wierszy podpisów jest zgodna z kolejnością wierszy wiadomości (pierwszy wiersz podpisów odpowiada pierwszej wiadomości, drugi — drugiej itd.)

Funkcja skrótu *skrot*(*wiadomość*)

Skrót wiadomości jest 8-znakowym napisem, złożonym z wielkich liter alfabetu angielskiego.

Aby go wyznaczyć, wykonaj następujące kroki:

a) Wpisz do 8-elementowej tablicy *S* kody ASCII znaków słowa ”ALGORYTM”.
b) Treść wiadomości uzupełnij na końcu znakami kropki ’.’ do wielokrotności 8 znaków.
c) Rozpatrz po kolei 8-znakowe porcje treści wiadomości. W zależności od kodów ich znaków aktualizuj wartości elementów w tablicy S. Dla każdej porcji treści wiadomości powtarzaj:
   dla *j* =1,2 .. 8 wykonuj
   S[*j*] ← (*S*[*j*]) + kod znaku na *j*-tej pozycji w bieżącej porcji wiadomości) *mod* 128

d) Zbuduj wynik, wyznaczając jego kolejne znaki na podstawie elementów tablicy S:
   wynik = ””
   dla *j* =1,2 .. 8 wykonuj
   wynik ← *wynik* + char(65 + *S*[*j*] *mod* 26)

   gdzie: *mod* jest operatorem dzielenia modulo,
   funkcja *char*(*kod*) zwraca reprezentację graficzną znaku o podanym kodzie

Otrzymany w ten sposób *wynik* jest skrótem wiadomości.

Algorytm *A* szyfrowania z kluczem prywatnym (*e*,*n*) i deszyfrowania kluczem publicznym (*d*,*n*)

Deszyfrowanie polega na wykonaniu operacji *x=(y*d mod n)*, gdzie za *y* należy przyjąć kolejne liczby tworzące podpis elektroniczny. Tekst wynikowy można otrzymać, składając w jeden napis reprezentacje graficzne kolejnych liczb *x* zgodnie ze standardem ASCII.

Uwaga dla dociekliwych

Zaszyfrowanie algorytmem *A* polegało na wykonywaniu operacji ***y=(x*e mod n)***, gdzie za *x* należało podstawić kody ASCII kolejnych znaków tekstu źródłowego. Uzyskany w ten sposób ciąg liczb jest podpisem elektronicznym wiadomości. Gdyby ktoś chciał złamać szyfr *A*, czyli wyznaczyć nieznany element *e* klucza prywatnego, musiałby znaleźć taką wartość *e,* względnie pierwszą z *d*, że *e*d mod n = 1*. Uzasadnienia szukaj w prawach arytmetyki modularnej.

Napisz program rozwiązujący poniższe zadania. Do oceny oddaj plik tekstowy `epodpis_wynik.txt`, zawierający odpowiedzi, oraz plik (pliki) zawierający reprezentację komputerową Twojego rozwiązania.

78.1.

Wyznacz skrót **pierwszej** wiadomości z pliku `wiadomosci.txt` i udokumentuj wyniki kolejnych etapów obliczania tego skrótu. Zapisz w kolejnych wierszach pliku wynikowego:

a) liczbę znaków wiadomości po jej uzupełnieniu do najmniejszej długości o wielokrotności 8 znaków,

b) wartości liczbowe 8 kolejnych bajtów skrótu (elementy tablicy *S*) po przetworzeniu całej wiadomości — wszystkie wartości w jednym wierszu, oddzielone pojedynczymi znakami odstępu,

c) skrót wiadomości w postaci napisu o długości 8, złożonego z wielkich liter alfabetu angielskiego.

78.2.

Odszyfruj skróty wiadomości ze wszystkich podpisów elektronicznych umieszczonych w pliku `podpisy.txt`, stosując algorytm *A* z kluczem publicznym (*d*,*n*) = (3,200). Zapisz uzyskane skróty w kolejnych, osobnych wierszach pliku z odpowiedziami.

78.3.

Zweryfikuj wiarygodność wszystkich wiadomości i podaj numery wiadomości wiarygodnych. Zapisz w jednym wierszu pliku z odpowiedziami, jako liczby z zakresu 1..11, zgodnie z kolejnością umieszczenia ich w pliku danych, oddzielone pojedynczym znakiem odstępu.

Zadanie 79.
Wiązka zadań *Okręgi*

Okrąg na płaszczyźnie reprezentujemy za pomocą trzech liczb x, y, r, gdzie (x, y) oznaczają współrzędne środka okręgu, a r — jego promień.

W pliku `okregi.txt` danych jest 2000 okręgów; każdy zapisany jest w osobnym wierszu, zawierającym trzy liczby rzeczywiste x, y, r $(-10^9 \leq x, y \leq 10^9,\ 0 < r < 10^9)$.

Pierwsze **dziewięć wierszy** tego pliku zawiera liczby:

```
  8.000          5.000         40.815
 22.623         21.558         50.000
 53.551         38.508         25.663
 63.429         42.655         18.844
 77.352         48.799         20.286
 80.274         52.627         16.127
178.318        128.106         23.993
191.501        140.711          8.379
197.073        143.275         13.129
```

Można zauważyć, że ostatnie siedem okręgów w całości zawiera się w pierwszej ćwiartce układu współrzędnych. Natomiast pierwsze dwa okręgi nie leżą w całości w żadnej ćwiartce.

Napisz program, który umożliwi wykonanie poniższych poleceń. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym `wyniki_okregi.txt`. Wyniki do każdego zadania poprzedź numerem je oznaczającym.

79.1.

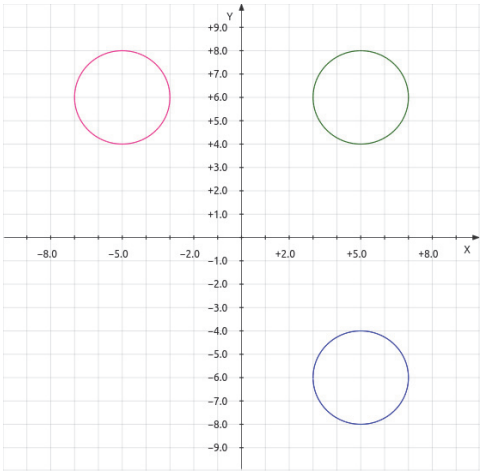
Podaj liczbę okręgów, które całkowicie zawierają się w I, II, III i IV ćwiartce układu współrzędnych. Podaj również liczbę okręgów, które nie zawierają się w całości w żadnej ćwiartce, tzn. mają co najmniej jeden punkt wspólny z jedną z osi Ox lub Oy. Jako odpowiedź wypisz pięć liczb: liczba okręgów I ćwiartki, liczba okręgów II ćwiartki, liczba okręgów III ćwiartki, liczba okręgów IV ćwiartki oraz liczba okręgów, które nie zawierają się w całości w żadnej ćwiartce.

79.2.

Powiemy, że dwa okręgi tworzą *lustrzaną parę*, jeśli jeden z nich powstaje przez odbicie drugiego względem jednej z osi Ox lub Oy. Podaj liczbę lustrzanych par spośród wszystkich okręgów zapisanych w pliku `okregi.txt`.

Uwagi

1. Układ trzech okręgów powstałych przez odbicia względem osi Ox lub Oy zawiera **dwie** lustrzane pary; np. wśród okręgów o środkach w punktach (-5,6), (5,6), (5,-6) (i o tych samych promieniach) są dokładnie dwie lustrzane pary; zob. poniższy rysunek.

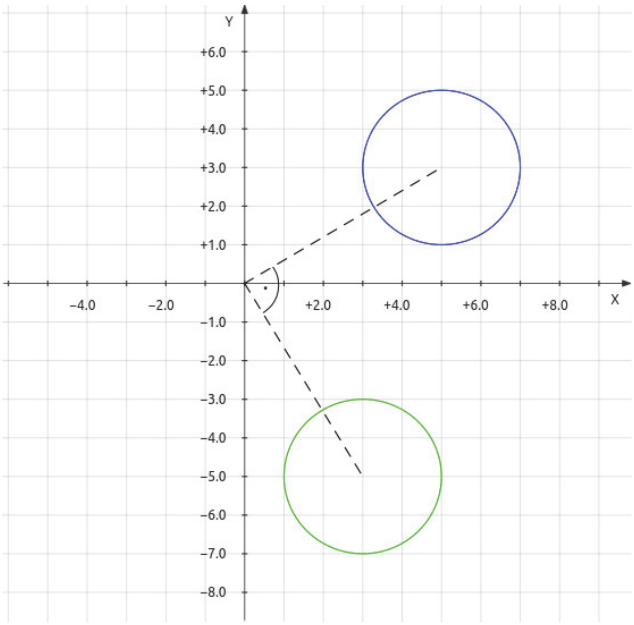


2. Analogicznie do poprzedniego punktu układ czterech okręgów zawiera **cztery** lustrzane pary.

79.3.

Powiemy, że dwa okręgi tworzą *prostopadłą parę*, jeśli jeden z nich powstaje przez obrót drugiego o 90 stopni względem środka układu współrzędnych.

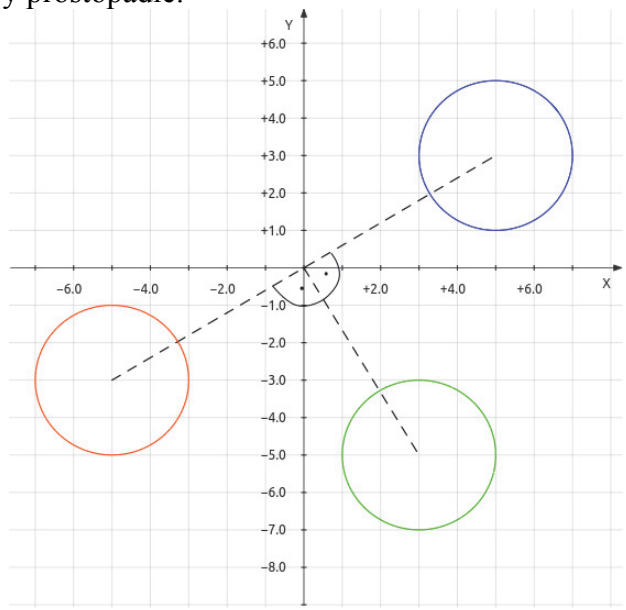
Przykład: okręgi o środkach w punktach (3,-5), (5,3) (i o tych samych promieniach) tworzą parę prostopadłą; zob. rysunek.



Podaj liczbę prostopadłych par okręgów spośród wszystkich okręgów zapisanych w pliku `okregi.txt`.

Uwagi

1. układ trzech okręgów powstałych przez obrót o 90 stopni względem środka układu współrzędnych zawiera **dwie** pary prostopadłe;
   np. wśród okręgów o środkach (5,3), (3,-5), (-5,-3) (i o tych samych promieniach) są dokładnie dwie pary prostopadłe.



2. Analogicznie do poprzedniego punktu układ czterech okręgów zawiera **cztery** pary prostopadłe.

79.4.

Powiemy, że ciąg okręgów tworzy *łańcuch*, jeśli kolejne okręgi tego ciągu mają ze sobą co najmniej jeden punkt wspólny; przyjmujemy, że ciąg zawierający tylko jeden okrąg również tworzy łańcuch. Można zauważyć, że wśród podanych okręgów mamy dwa łańcuchy: pierwszy o długości 6, a drugi o długości 3.

Znajdź długości wszystkich łańcuchów tworzonych przez okręgi zapisane w wierszach o numerach **od 1 do 1000**. Podaj długość najdłuższego łańcucha.

Zadanie 80.

Wiązka zadań *Trójkąty*

W pliku `dane_trojkaty.txt` znajduje się 500 liczb całkowitych dodatnich. Każda liczba jest zapisana w osobnym wierszu, żadna liczba **nie występuje w pliku więcej niż jeden raz**. Liczby podane w pliku `dane_trojkaty.txt` to długości odcinków, z których będziemy próbować budować trójkąty.

Napisz program(-y), który pozwoli rozwiązać poniższe zadania. Odpowiedzi zapisz w pliku `wyniki_trojkaty.txt`. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją numerem zadania.

80.1.

Wypisz wszystkie trójki **kolejnych** liczb z pliku `dane_trojkaty.txt`, które są długościami boków trójkąta prostokątnego.

Przykład

Rozważmy plik składający się z dziesięciu liczb: 8, 7, 4, 3, 5, 9, 12, 13, 85, 84. Wynikiem są dwie trójki liczb: 4, 3, 5 oraz 13, 85, 84. Trójkąt prostokątny tworzą też odcinki o bokach 5, 12 i 13, ale liczby te **nie są kolejnymi** liczbami w podanym pliku.

80.2.

Podaj największy obwód trójkąta, którego boki mają długości równe liczbom występującym w różnych wierszach pliku `dane_trojkaty.txt`.

Przykład

Dla pliku składającego się z dziesięciu liczb: 10, 18, 70, 100, 15, 13, 21, 12, 1, 2 wynikiem jest 54, ponieważ trójkąt o największym obwodzie ma boki 18, 15 i 21.

80.3.

Podaj, ile **nieprzystających** trójkątów można utworzyć z odcinków o długościach podanych w pliku `dane_trojkaty.txt`.

Uwaga: Dwa trójkąty są przystające wtedy i tylko wtedy, gdy trzy boki jednego trójkąta są odpowiednio równe trzem bokom drugiego trójkąta, np. trójkąt o bokach (10, 18, 15) jest przystający z trójkątem o bokach (18, 15, 10).

Przykład

Dla pliku składającego się z dziesięciu liczb: 10, 18, 70, 100, 15, 13, 21, 12, 1, 2 wynikiem jest 21, gdyż z podanych długości odcinków można utworzyć 21 trójkątów o następujących bokach:

(10, 18, 15); (10, 18, 13); (10, 18, 21); (10, 18, 12); (10, 15, 13); (10, 15, 21); (10, 15, 12); (10, 13, 21); (10, 13, 12); (10, 21, 12); (18, 15, 13); (18, 15, 21); (18, 15, 12); (18, 13, 21); (18, 13, 12); (15, 13, 21); (15, 13, 12); (15, 21, 12); (13, 21, 12); (13, 12, 2); (18, 21, 12).

Zadanie 81.

Wiązka zadań *Czworokąty*

Plik `wierzcholki.txt` zawiera 100 wierszy. W każdym wierszu zapisano 6 liczb całkowitych z przedziału <–100; 100>, będących współrzędnymi trzech różnych punktów: A, B i C w kartezjańskim układzie współrzędnych (odpowiednio x_a, y_a, x_b, y_b, x_c, y_c). Liczby w wierszu są oddzielone pojedynczymi znakami tabulacji.

Podobny plik `wierzcholkiTR.txt` zawiera również 100 wierszy. W każdym wierszu zapisano 6 liczb całkowitych należących do przedziału <–100; 100>, będących współrzędnymi trzech wierzchołków trójkąta ABC (odpowiednio x_a, y_a, x_b, y_b, x_c, y_c). Liczby w wierszu są oddzielone pojedynczymi znakami tabulacji.

Wykorzystując dane zawarte w plikach oraz dostępne narzędzia informatyczne, rozwiąż poniższe zadania.

Napisz program(-y), za pomocą którego(-ych) uzyskasz odpowiedzi do poniższych zadań. Do oceny oddaj dokument wyniki.txt z zapisanymi odpowiedziami na poszczególne zadania oraz pliki źródłowe programów wykorzystanych do uzyskania rozwiązania.

81.1.

Podaj liczbę wierszy z pliku `wierzcholki.txt`, w których wszystkie zapisane punkty leżą w I ćwiartce układu współrzędnych i nie należą do osi OX i OY.

81.2.

Podaj liczbę wierszy z pliku `wierzcholki.txt`, w których zapisane są współrzędne punktów leżących na jednej prostej.

81.3.

Podaj (z pliku `wierzcholkiTR.txt`) współrzędne wierzchołków trójkąta o największym obwodzie oraz obwód tego trójkąta. Obwód zaokrąglij do dwóch miejsc po przecinku. Uwaga: możesz założyć, że jest tylko jeden taki trójkąt.

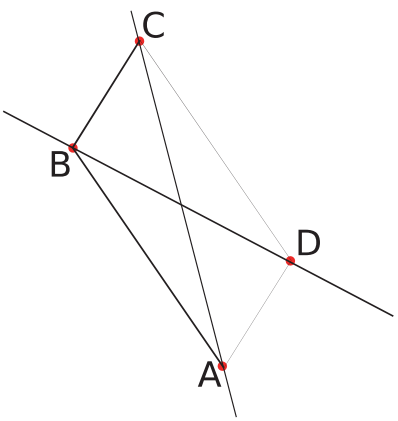
79.4.

Dla każdego wiersza z pliku `wierzcholkiTR.txt` sprawdź, czy punkty zapisane w tym wierszu są wierzchołkami pewnego trójkąta prostokątnego. Podaj liczbę trójkątów prostokątnych zapisanych w tym pliku

Uwaga: Takich trójkątów jest więcej niż cztery.

81.5.

Dla każdego wiersza z pliku `wierzcholkiTR.txt`, w których zapisane są kolejno współrzędne punktów A, B i C, wyznacz współrzędne punktu D, tak aby czworokąt ABCD był równoległobokiem. Podaj współrzędne wszystkich wierzchołków czworokątów ABCD, których punkt D leży na prostej $y=x$.

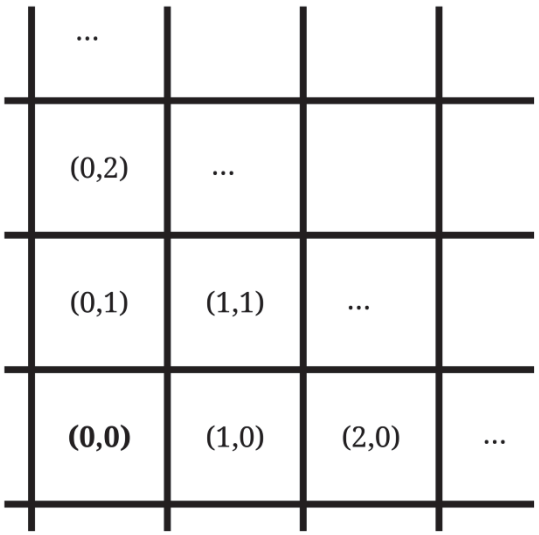


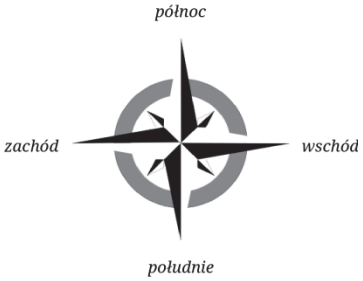
2.2. Symulacja

Zadanie 82.

Wiązka zadań *Piraci*

Na karaibskiej wyspie Santo Domingo piraci poszukują skarbu, który niegdyś ukrył tam zbuntowany hiszpański wicekról. Przeszukiwany teren podzielili na jednostkowe kwadratowe obszary o boku jednej mili. Każdemu obszarowi przypisali współrzędne: odciętą i rzędną, przy czym odcięta rośnie w kierunku wschodnim, a rzędna w kierunku północnym, tak jak przedstawiono na poniższym rysunku:





W środę 1 października 1902 roku piraci lądują na wyspie, na obszarze (0,0), i postępują według następującej, znalezionej w tajemniczych okolicznościach, instrukcji:

> *„Każdego dnia przejdź 8 mil prosto na północ, a potem skręć w prawo i przejdź 11 mil na wschód. Policz, ile w sumie mil na północ przeszedłeś od zejścia ze statku — kopiąc w tym miejscu, znajdziesz tyle złotych dublonów, ile cyfr ma ta łączna odległość. Ponadto każdego trzeciego dnia miesiąca znajdziesz dodatkowo dwa dublony.*
>
> *Po zebraniu złota zawróć i przejdź na zachód tyle mil, ile dublonów właśnie zebrałeś. Tam rozbij obóz na noc, a następnego dnia możesz kontynuować swoją wędrówkę.*
>
> *Nie bądź zbyt chciwy, aby nie spadła na ciebie klątwa!”*

Wiedząc, że piraci spędzili na wyspie 150 dni, każdego dnia (łącznie z dniem lądowania) wypełniając dokładnie te polecenia. Jeżeli piraci znajdują się w kwadracie o współrzędnych (i,j), to po przejściu 1 mili na północ znajdą się w obszarze o współrzędnych $(i,j+1)$. Z kolei w obszarze o współrzędnych $(i+1,j)$ znajdą się, gdy pójdą na wschód, $(i,j-1)$ — gdy pójdą na południe, $(i-1,j)$ — gdy pójdą na zachód.

Używając dostępnych narzędzi informatycznych, znajdź odpowiedzi na poniższe pytania. Odpowiedzi zapisz w pliku o nazwie *wyniki.txt,* każdą z nich umieszczając w osobnym wierszu i poprzedzając numerem odpowiedniego zadania.

82.1.

Codziennie wieczorem piraci obozują w miejscu, gdzie znaleźli się po wykopaniu dublonów i przejściu odpowiedniej liczby mil na zachód. Podaj współrzędne obozu piratów, w którym spędzają noc wigilijną 24/25 grudnia 1902.

82.2.

Oblicz, ile mil łącznie przejdą piraci przez cały okres poszukiwań. Uwzględnij mile przebyte na północ, wschód, a także na zachód, w czasie cofania się.

82.3.

W każdą sobotę część piratów wymyka się z obozu, aby popłynąć łódką na sąsiednią wyspę Tortuga, gdzie na różne rozrywki tracą 10% (zaokrąglone w dół do liczby całkowitej) majątku posiadanego przez całą bandę.

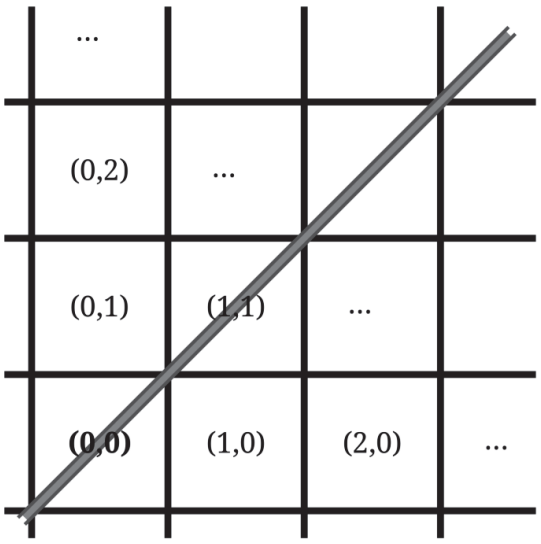
Oblicz, ile łącznie dublonów piraci zostawią na Tortudze przez cały okres swojej wyprawy. Załóż, że na wyspę przybyli, nie mając ani jednego dublona.

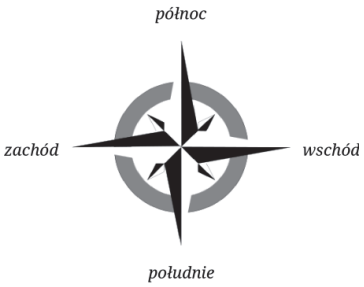
82.4.

Klątwa uaktywnia się (o czym piraci nie wiedzą) każdego dnia, w którym piraci znajdują 4 lub więcej dublonów. Klątwa ta skutkuje nieprzyjemną niespodzianką: po powrocie na statek piraci będą musieli zmierzyć się z oddziałem wojska miejscowego gubernatora, składającym się z tylu żołnierzy, ile piraci zebrali łącznie dublonów w feralne dni. Oblicz, ilu przeciwników spotkają piraci.

82.5.

Przez Santo Domingo płynie rzeka, pod kątem 45 stopni do brzegów wyspy, wpadająca do morza w pobliżu punktu lądowania piratów. Odległość obozu piratów (będącego w danym dniu na polu (x,y) od rzeki można opisać wzorem *odległość* = $|x - y|$, czyli jako wartość bezwzględną z różnicy współrzędnych obozu[1].





[1] W rzeczywistości odległość ta wynosi $\frac{|x-y|}{\sqrt{2}/2}$ (licząc od środka obszaru do rzeki), użyjemy jednak uproszczonej formuły.

Znajdź średnią odległość wieczornego obozu piratów od rzeki przez cały okres poszukiwań (150 dni) w zaokrągleniu do dwóch miejsc po przecinku, a następnie sporządź wykres kolumnowy przedstawiający tę odległość w kolejnych dniach.

Zadanie 83.

Wiązka zadań Wilki i zające

Klasyczny model Lotki-Volterry opisuje interakcje między populacjami dwóch gatunków. Niech W_n i Z_n oznaczają liczebność populacji odpowiednio wilków i zajęcy, gdzie n jest numerem kolejnego miesiąca. Model definiują następujące dwa wzory:

$$Z_{n+1} = Z_n + a\,Z_n - b\,Z_n W_n,$$
$$W_{n+1} = W_n + b\,Z_n W_n - c\,W_n,$$

gdzie a, b, c są parametrami modelu i mają następujące znaczenie:

- a — współczynnik przyrostu liczby zajęcy,
- b — współczynnik umierania zajęcy na skutek polowań wilków,
- c — współczynnik umierania wilków.

Przyjmujemy następujące początkowe wartości populacji:

$$Z_0 = 100, \qquad W_0 = 30.$$

Rozważamy model Lotki-Volterry z parametrami:

$$a = 0{,}02, \qquad b = 0{,}0005, \qquad c = 0{,}05.$$

Dla potrzeb matematycznych modeli symulacyjnych przyjmujemy, że wartości Z_n i W_n mogą być liczbami niecałkowitymi. Poniżej podano wartości populacji wilków i zajęcy w pierwszych 5 miesiącach symulacji (wartości zaokrąglono do 2 miejsc po przecinku):

n	**Zające** (Z_n)	**Wilki** (W_n)
0	100,00	30,00
1	100,50	30,00
2	101,00	30,01
3	101,51	30,02
4	102,01	30,05
5	102,52	30,08

Wykorzystując dostępne narzędzia informatyczne, wykonaj poniższe zadania. Odpowiedzi (z wyjątkiem wykresu do zadania 3) zapisz w pliku o nazwie `wyniki_wilki.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 3 umieść w pliku `wykres_wilki.xxx`, gdzie *xxx* oznacza rozszerzenie odpowiednie dla formatu pliku.

83.1.

Podaj liczebność populacji wilków i zajęcy po upływie 5 lat (60 miesięcy). Wynik podaj z dokładnością do 2 miejsc po przecinku.

83.2.

Dla podanych powyżej wartości *a, b, c* i wartości początkowych Z_0, W_0 przebieg symulacji jest następujący:

- na początku następuje przyrost obu populacji;
- gdy wilków przybędzie odpowiednio dużo, to populacja zajęcy zaczyna maleć;
- gdy liczebność populacji zajęcy zaczyna spadać, to zaczyna też spadać liczebność populacji wilków.

Podaj, kiedy zacznie maleć populacja zajęcy, tzn. podaj najmniejszą wartość n, dla której $Z_n < Z_{n-1}$. Podaj też, kiedy nastąpi spadek populacji wilków, tzn. podaj najmniejszą wartość m, dla której $W_m < W_{m-1}$.

83.3.

Utwórz wykres liniowy przedstawiający liczebność populacji wilków i zajęcy w kolejnych miesiącach w ciągu pierwszych 20 lat symulacji. Pamiętaj o czytelnym opisie wykresu.

83.4.

Rozważ symulację liczebności populacji wilków i zajęcy w okresie 40 lat. Podaj najmniejszą i największą liczebność wilków i zajęcy, jaka może być wskazana w rozważanym modelu. Odpowiedzi podaj z dokładnością do 2 miejsc po przecinku.

	Zające	Wilki
Najmniejsza liczebność populacji		
Największa liczebność populacji		

Zadanie 84.

Wiązka zadań *LPG*

Pan Binarny założył instalację gazową w swoim samochodzie, dzięki czemu mógł wykorzystywać do jazdy 2 rodzaje paliwa. Parametry pojazdu są przedstawione w poniższej tabeli:

paliwo	**pojemność zbiornika**	**zużycie paliwa (spalanie)**
Pb95	45 litrów	6 litrów na 100 km
LPG	30 litrów	9 litrów na 100 km

Pan Binarny postanowił prowadzić rejestr długości pokonywanych tras w kolejnych dniach 2014 roku. Dane te zostały zebrane w pliku `lpg.txt`. Są to data wyjazdu (`data`) oraz liczba kilometrów przejechanych tego dnia (`km`). Pierwszy wiersz jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

Przykład

```
data          km
2014-01-01    159
2014-01-02    82
2014-01-03    108
```

1 stycznia 2014 r. zbiorniki z paliwem (Pb95 i LPG) były zatankowane do pełna. Codziennie rano pan Binarny decydował, w jaki sposób będzie zużywał paliwo. Jeżeli w zbiorniku LPG było więcej niż 15 litrów, to do jazdy wykorzystywał tylko paliwo LPG, w przeciwnym razie połowę trasy pokonywał, korzystając z paliwa Pb95, a połowę — z paliwa LPG.

Liczbę zużytych litrów paliwa na trasie obliczał ze wzoru

$$litry = \frac{spalanie * km}{100},$$

gdzie *spalanie* oznacza zużycie paliwa w litrach na 100 km (zgodnie z parametrami zużycia paliwa w tabeli powyżej), a *km* oznacza liczbę przejechanych kilometrów, przy czym wyniki **zaokrąglał do dwóch miejsc po przecinku**.

Bez względu na rodzaj paliwa tankowanie wykonywane było **po przejechaniu trasy, wieczorem**. **W każdy czwartek**, o ile w zbiorniku z Pb95 znajdowało się poniżej 40 litrów, pan

Binarny tankował do pełna Pb95. Paliwo LPG było tankowane do pełna wtedy, gdy w zbiorniku LPG znajdowało się mniej niż 5 litrów tego paliwa i odbywało się **niezależnie od dnia tygodnia**.

Wykorzystując dane zawarte w pliku `lpg.txt` i dostępne narzędzia informatyczne, wykonaj poniższe zadania. Odpowiedzi (za wyjątkiem wykresu do zadania 3) zapisz w pliku o nazwie `wyniki_lpg.txt`. Wyniki do każdego zadania poprzedź numerem je oznaczającym. Wykres do zadania 3 umieść w pliku `wykres_lpg.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku.

Uwaga: Dane podane w pliku `lpg.txt` wykluczają sytuację, że panu Binarnemu kiedykolwiek zabrakło paliwa na trasie i w związku z tym musiał on zmieniać decyzję co do rodzaju używanego paliwa.

84.1.

Podaj, ile razy pan Binarny tankował paliwo Pb95, a ile razy LPG.

Podaj liczbę dni, w których pan Binarny korzystał podczas jazdy wyłącznie z paliwa LPG.

84.2.

Podaj dzień, w którym w zbiorniku LPG po raz pierwszy było rano mniej niż 5,25 litra paliwa.

84.3.

Utwórz wykres, w którym porównasz stan zbiornika LPG przed podróżą ze stanem zbiornika LPG po podróży dla pierwszych 31 dni. Pamiętaj o czytelnym opisie wykresu.

84.4.

Przyjmijmy, że w całym rozważanym okresie koszt 1 litra LPG jest równy 2,29 zł, a koszt 1 litra Pb95 wynosi 4,99 zł. Koszt instalacji gazowej wynosi 1600 zł.

Oblicz dla rozważanego okresu koszt eksploatacji samochodu z wykorzystaniem instalacji gazowej (uwzględnij koszt instalacji oraz zużyte LPG) oraz koszt eksploatacji samochodu wykorzystującego tylko paliwo Pb95 (w oparciu o zużytą ilość Pb95).

W swoich obliczeniach dotyczących eksploatacji z wykorzystaniem instalacji gazowej przyjmij wszystkie opisane założenia dotyczące tankowania i sposobu przejazdów pana Binarnego. Zauważ, że w obliczeniach związanych z wykorzystaniem wyłącznie paliwa Pb95 nie ma znaczenia dzień tankowania, istotny jest jedynie roczny koszt eksploatacji.

Podaj oba koszty eksploatacji. Podczas obliczeń wszystkie koszty oraz wyniki końcowe zaokrąglaj do dwóch miejsc po przecinku.

Zadanie 85.

Wiązka zadań *Oscypki*

Oscypek to ser sezonowy. Robi się go z mleka owiec wypasanych na halach. Sezon wypasu trwa od 23 kwietnia (św.Wojciecha) do 29 września (św. Michała).

Baca ma stado owiec składające się z **600** sztuk. Każda owca przez pierwsze 7 dni sezonu daje **0,5 l** mleka dziennie. Począwszy od ósmego dnia (czyli 30.04) ta ilość mleka wzrasta skokowo o **4%** co **7** dni. Tak dzieje się do 24 czerwca (św.Jana) włącznie. Od 25 czerwca

co 7 dni ilość mleka od jednej owcy zmniejsza się skokowo o **10%** w stosunku do ilości mleka z poprzedniego 7-dniowego okresu. Proces ten trwa do końca wypasu owiec. Liczbę odnoszącą się do rzeczywistej ilości mleka oddawanego przez jedną owcę w każdym dniu, wyliczoną zgodnie z powyższymi regułami, należy zaokrąglić do dwóch miejsc po przecinku.

Ser jest wyrabiany z mleka zebranego od owiec poprzedniego dnia. Jego wyrób trwa jedną dobę, a następnie jest on wędzony przez 4 doby i kolejnego dnia jest gotowy do sprzedaży. Na wytworzenie jednego sera potrzeba **6** litrów mleka. Z mleka, jakie jest do dyspozycji, baca produkuje tylko całe sery i maksymalną możliwą ich liczbę. Reszta mleka, niezużyta w danym dniu do produkcji serów, jest wykorzystywana do innych celów.

Każdego dnia w okresie od 29.04 do 29.09 włącznie do bacówki przychodzą po sery turyści. W dni powszednie kupują 36 serów dziennie (o ile baca ma tyle na sprzedaż), a w dni weekendu (sobotę, niedzielę) — 100 serów dziennie. Jeśli danego dnia baca ma na sprzedaż mniej serów niż chcą kupić turyści, to wykupują je wszystkie, jeśli więcej, to baca przechowuje sery i ma je do sprzedania w kolejnych dniach. Rozważmy, jak wyglądało wyrabianie i sprzedaż serów w bacówce w 2014 roku.

Przykład

23.04 — pierwsze dojenie owiec,

24.04 — wyrób serów z mleka z dnia 23.04,

25.04 – 28.04 — wędzenie serów zrobionych 24.04,

29.04 — pierwsze sery gotowe do sprzedaży.

We wtorek 29.04.2014 do bacówki od rana przychodzili turyści i kupowali gotowe sery. Baca miał 50 gotowych serów, z których turyści kupili 36 sztuk.

Wykorzystując dostępne narzędzia informatyczne, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki.txt` (z wyjątkiem wykresu do zadania 4). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 4 umieść w pliku `wykres.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku z Twoimi rozwiązaniami.

85.1.

Podaj liczbę litrów mleka, jaką uzyskał baca ze swojego stada owiec w okresie od 23.04.2014 do 29.09.2014 włącznie.

85.2.

Podaj łączną liczbę serów, które turyści byli gotowi zakupić u bacy począwszy od dnia, kiedy sery były gotowe do sprzedaży, czyli od 29.04.2014, do ostatniego dnia wypasu, czyli do 29.09.2014.

85.3.

Podaj dzień, w którym po raz pierwszy w sezonie (od 29.04.2014 do 29.09.2014) baca miał mniej gotowych serów, niż chcieli kupić turyści. Ile było takich dni w całym sezonie?

85.4.

Wykonaj zestawienie, w którym dla każdego miesiąca, od kwietnia do września włącznie, podasz liczbę sprzedanych w tym miesiącu oscypków. Do wykonanego zestawienia utwórz wykres kolumnowy. Pamiętaj o czytelnym opisie wykresu.

85.5.

Podaj najmniejszą liczbę owiec, jaką musiałby posiadać baca, aby każdego dnia od 29 kwietnia do 29 września zaspokoić popyt turystów na oscypki.

85.6.

W przepisie na produkcję oscypka dopuszcza się dodawanie mleka krowiego do mleka owczego (maksymalniy udział mleka krowiego może wynieść 40%). Podaj liczbę serów (gotowych do sprzedaży), które można byłoby wyprodukować w całym okresie wypasu przy opisanych wyżej warunkach uzyskiwania mleka owczego, jeżeli ser tworzony byłby:

- tylko z mleka owczego,
- z mieszanki składającej się z 80% mleka owczego i 20% mleka krowiego,
- z mieszanki składającej się z 60% mleka owczego i 40% mleka krowiego.

Zadanie 86.

Wiązka zadań *Wybory*

Jedną z metod obsadzania mandatów w systemach wyborczych opartych na proporcjonalnej reprezentacji partii politycznych jest metoda Sainte-Laguë. W metodzie tej kolejne mandaty w danym okręgu wyborczym przydzielane są po kolei. Przed przydzieleniem każdego mandatu wyznacza się współczynnik w_K dla każdego komitetu wyborczego K równy

$$w_k = \frac{g_k}{2m_K+1},$$

gdzie:

g_k — liczba głosów zdobytych przez komitet K w rozważanym okręgu,

m_K — liczba mandatów przydzielonych komitetowi K do tej pory.

Kolejny mandat otrzymuje ten komitet wyborczy K, który ma największą wartość współczynnika w_k.

Uwaga. Jeśli kilka komitetów ma taką samą wartość współczynnika w_k. Sposób rozstrzygnięcia takich remisów nie ma wpływu na rozwiązania poniższych zadań, dlatego takiego przypadku nie opisujemy.

Przykład

Przyjmijmy, że do wyborów zgłosiły się dwa komitety wyborcze $K1$ i $K2$, które otrzymały odpowiednio $g_{K1} = 810$ oraz $g_{K2} = 300$ głosów w okręgu, w którym do obsadzenia są 4 mandaty.

Mandat pierwszy otrzymuje $K1$, gdyż $m_{K1} = 0$, $m_{K2} = 0$ oraz:

$$w_{K1} = \frac{810}{1} = 810,\ w_{K2} = \frac{300}{1} = 300.$$

Mandat drugi otrzymuje $K2$, gdyż $m_{K1} = 1$, $m_{K2} = 0$ oraz:

$$w_{K1} = \frac{810}{3} = 270, w_{K2} = \frac{300}{1} = 300.$$

Mandat trzeci otrzymuje $K1$, gdyż $m_{K1} = 1, m_{K2} = 1$ oraz:

$$w_{K1} = \frac{810}{3} = 270, w_{K2} = \frac{300}{3} = 100.$$

Mandat czwarty również otrzymuje $K1$, gdyż $m_{K1} = 2, m_{K2} = 1$ oraz:

$$w_{K1} = \frac{810}{5} = 162, w_{K2} = \frac{300}{3} = 100.$$

W rezultacie komitet $K1$ otrzyma 3 mandaty, a komitet $K2$ otrzyma 1 mandat.

Wybory odbyły się w 20 okręgach wyborczych, zgłosiło się do nich pięć komitetów wyborczych o nazwach $K1$, $K2$, $K3$, $K4$ i $K5$. Do obsadzenia jest 400 mandatów poselskich. Plik `dane_wybory.txt` zawiera wyniki przeprowadzonych wyborów. Każdy wiersz pliku zawiera dane dotyczące jednego okręgu wyborczego: nazwę okręgu i 5 liczb oznaczających liczbę głosów oddanych na kolejne komitety wyborcze ($K1$, $K2$, $K3$, $K4$ i $K5$) w tym okręgu (zakładamy, że każdy głos wskazuje na jeden komitet, nie ma głosów nieważnych). Wartości w wierszu oddzielone są spacjami.

Przykład

```
A1 16573 13009 19177 21253 9656
B2 24574 10394 9756 13299 3464
```

Wykorzystując dane zawarte w tym pliku i dostępne narzędzia informatyczne, wykonaj poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_wybory.txt` (z wyjątkiem wykresu do zadania 1). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 1 umieść w pliku `wyniki_wybory.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku.

86.1.

Podaj, ile ogółem głosów oddano (na wszystkie komitety razem) w każdym z okręgów wyborczych. Na tej podstawie sporządź wykres kolumnowy porównujący liczby głosów oddanych w poszczególnych okręgach. Pamiętaj o prawidłowym i czytelnym opisie wykresu.

86.2.

Poparcie komitetu wyborczego w danym okręgu to procent głosów zdobytych w tym okręgu, w zaokrągleniu do dwóch miejsc po przecinku. *Matecznikiem* komitetu wyborczego nazywamy okręg, w którym komitet ten zdobył największe procentowe poparcie w porównaniu z jego poparciem w innych okręgach. Dla każdego komitetu wyborczego podaj nazwę okręgu będącego jego *matecznikiem*.

Przykład

Przyjmijmy, że wybory odbyły się tylko w dwóch okręgach wyborczych $A1$ i $B2$, z następującymi wynikami:

	$K1$	$K2$	$K3$	$K4$	$K5$
$A1$	100	300	400	50	150
$B2$	150	500	750	400	200

Wówczas matecznikiem komitetu $K1$ jest okręg $A1$ (100 głosów, poparcie 10%), mimo że uzyskał on większą liczbę głosów w okręgu $B2$ (150 głosów, poparcie 7,5%).

86.3.

Przyjmijmy, że w każdym okręgu wyborczym jest do obsadzenia 20 mandatów, przy czym przyznawane są one w oparciu o rozkład głosów w tym okręgu zgodnie z metodą Sainte-Laguë. Dla każdego z komitetów wyborczych ($K1$, $K2$, $K3$, $K4$, $K5$) wskaż maksymalną liczbę mandatów zdobytych przez ten komitet w jednym okręgu.

86.4.

Pierwsza litera nazwy każdego okręgu oznacza region, w którym ten okręg się znajduje: A, B, C lub D. Komisja wyborcza rozważa różne warianty podziału mandatów:

a) standardowy: w każdym okręgu mandaty obsadzane są osobno, przyznaje się po 20 mandatów na okręg metodą Sainte-Laguë.

b) regionalny: wszystkie okręgi z tego samego regionu łączone są w duży okręg i w tak utworzonych większych okręgach (A, B, C i D) obsadza się po 100 mandatów metodą Sainte-Laguë.

Podaj liczbę mandatów zdobytych w obu wariantach przez każdy komitet w całym kraju.

86.5.

Rozważmy sytuację, w której o mandaty rywalizują dwa komitety wyborcze, Q i R, w jednym okręgu wyborczym, w którym każdy spośród 100 000 wyborców oddaje głos na komitet Q albo komitet R. Do obsadzenia jest $2m$ mandatów. Ustal najmniejszą liczbę głosów, jakie musi zdobyć komitet Q, aby zdobyć dokładnie połowę, czyli m mandatów, dla m=10, m=20 i m=50.

Zadanie 87.

Wiązka zadań *U-977*



Okręt podwodny U-977 wypłynął w swój pierwszy patrol bojowy z Kristiansand w czwartek 3 maja 1945 o godzinie 00:00 z zadaniem dotarcia do Southampton, a celem wyprawy było zatopienie jak największej liczby jednostek alianckich. W czasie rejsu na okręt dotarła wiadomość o kapitulacji Niemiec. W tej sytuacji dowódca wraz z większością załogi zdecydowali się płynąć do Argentyny i rozpoczęli długi rejs na południową półkulę.

Ze względu na trudności w podróży 14 lipca 1945, tj. 73. dnia podróży, załoga postanowiła się zatrzymać na 24-godzinny odpoczynek w pobliżu archipelagu Wysp Zielonego Przylądka (14 lipca 1945, od godziny 0:00 do godziny 24:00). Mimo że w tym czasie okręt stał w miejscu, to wszystkie jego podzespoły pracowały, co powodowało ich naturalny proces zużywania się.

18 sierpnia 1945 o godzinie 00:00, po 107 dniach podróży, U-977 zawinął do Mar del Plata, gdzie załoga poddała się argentyńskiej marynarce.

Okręt był wyposażony w silniki elektryczne i akumulatory umożliwiające rejs w zanurzeniu, początkowo z prędkością średnią 4 węzłów (4 mile morskie na godzinę) oraz silniki spalinowe, umożliwiające rejs na powierzchni początkowo z prędkością średnią 10 węzłów.

Pojemność akumulatorów wystarczała na maksymalnie 20 godzin pracy silników elektrycznych. Podczas rejsu na silnikach spalinowych jednocześnie ładowane były akumulatory silników elektrycznych. Pełne naładowanie akumulatorów następowało po czterech godzinach używania silników spalinowych, niezależnie od prędkości.

Aby zmniejszyć prawdopodobieństwo wykrycia okrętu przez siły alianckie o ile było to możliwe okręt płynął w zanurzeniu korzystając z silników elektrycznych.

W związku z różnego rodzaju awariami i ogólnym zużyciem mechanizmów prędkość średnia na każdym z napędów zmniejszała się codziennie o 1%, niezależnie od czasu ich wykorzystywania. Począwszy od 15 lipca, w związku z uszkodzeniem akumulatorów, okręt płynął codziennie po 11 godzin w wynurzeniu. Pozostałą część doby przebywał w zanurzeniu, nie zmieniając pozycji nad dnem. Ostatniej doby rejsu, w związku z trudnościami nawigacyjnymi okręt płynął tylko 9 godzin.

Wszystkie określenia godzin w zadaniu wyrażone niezależnie od miejsca zdarzenia podane są według czasu uniwersalnego (GMT).

Wykorzystując dostępne narzędzia informatyczne, rozwiąż podane poniżej zadania. Obliczenia wykonuj na liczbach rzeczywistych i nie zaokrąglaj wyników częściowych. Do oceny oddaj dokument `wyniki.txt` z zapisanymi odpowiedziami do poszczególnych zadań oraz plik zawierający obliczenia.

87.1.

Którego dnia podróży średnia prędkość rejsu w zanurzeniu po raz pierwszy wyniosła poniżej 3 węzłów?

87.2.

Jaką drogę (w milach morskich) przebył okręt na trasie z Kristiansand do Mar del Plata? Odpowiedź podaj w zaokrągleniu do jednego miejsca po przecinku.

87.3.

Oblicz drogę (w milach morskich) przebytą przez okręt każdego dnia i sporządź wykres ilustrujący odległość przebytą w kolejnych dniach rejsu. Zadbaj o czytelny opis wykresu.

87.4.

Którego dnia zakończyłby się rejs okrętu, gdyby po 14 lipca okręt płynął już ciągle dokładnie tym samym kursem i będąc w zanurzeniu utrzymywał prędkość według wcześniejszego harmonogramu (20 godzin w zanurzeniu na silnikach elektrycznych i 4 godziny na powierzchni na silnikach spalinowych)?

87.5.

Dokładnie w tym samym czasie co U-977 (tj. 3 maja 1945 o godzinie 00:00) z Kristiansand wypłynął też okręt U-997 o tych samych parametrach technicznych. Jego dowódca, podobnie jak dowódca U-977, postanowił płynąć do Ameryki Południowej. Mając świadomość przeszkód/problemów, jakie mogą nastąpić w podróży dowódca postanowił płynąć z prędkością

równą 90% maksymalnej oraz w każdą niedzielę, poza pierwszą, będąc w zanurzeniu, wyłączyć na dwie godziny wszystkie silniki i dokonać dokładnego ich przeglądu oraz konserwacji. Dzięki temu ubytek uzyskiwanej prędkości wynosił tylko 0,3% na dobę na każdym z napędów i nie trzeba było robić 24-godzinnej przerwy w podróży. Można było, bowiem przez cały czas poza dniami konserwacji płynąć po 20 godzin na dobę w zanurzeniu, a przez 4 godziny płynąć na powierzchni i ładować akumulatory. Okręt płynął inną drogą, ale również zakończył swój rejs 17 sierpnia 1945 o północy. Podaj, w zaokrągleniu do jednego miejsca po przecinku, jaką odległość przebył okręt do 17 sierpnia 1945, do godziny 24:00.

Zadanie 88.

Wiązka zadań *Sprzedaż choinek*

Firma produkcyjno-handlowa „Świerk" sprzedaje choinki przed Bożym Narodzeniem. Co roku rozpoczyna ich sprzedaż pierwszego i kończy dwudziestego czwartego grudnia. Firma dysponuje jednym samochodem przeznaczonym do transportu choinek. Samochód wieczorem wyjeżdża z plantacji i dojeżdża na plac sprzedaży rano przed jej rozpoczęciem. W ciągu dnia wraca na plantację.

Zaobserwowano, że ze względu na zainteresowanie klientów firma może sprzedać danego dnia liczbę choinek wyznaczoną wg wzoru:

$$L = \lfloor(-d^2 + 40d + 50)/10\rfloor,$$

gdzie d oznacza numer dnia sprzedaży (1 grudnia jest 1. dniem sprzedaży, a 24 grudnia jest 24. dniem sprzedaży). Jeśli na placu jest co najmniej L choinek, to L choinek zostaje sprzedanych. Gdy na placu znajduje się mniej drzewek, niż klienci gotowi byliby kupić (mniej niż L), to część klientów wybierała inne punkty sprzedaży, gdzie wybór drzewek był większy. W takim dniu firmie udaje się maksymalnie sprzedać zaokrąglone w dół do liczby całkowitej 90% choinek, które znalazły się na placu rano po przyjęciu dostawy. Firma zaplanowała dostawy po 50 drzewek w następujących dniach: pierwszego grudnia, czwartego grudnia oraz następnie w każdy dzień o numerze parzystym aż do 24 grudnia włącznie.

Wykorzystując dostępne narzędzia informatyczne, rozwiąż następujące zadania. Do oceny oddaj dokument `wyniki.txt` z odpowiedziami do poszczególnych zadań oraz plik z przeprowadzonymi obliczeniami.

88.1.

Ile choinek zostanie w sumie sprzedanych w całym okresie sprzedaży i ile choinek pozostanie na placu wieczorem 24 grudnia?

88.2.

Wykonaj wykres kolumnowy przedstawiający dla każdego dnia sprzedaży liczbę choinek które znajdowały się tego dnia rano na placu (już po dostawie), z podziałem na sprzedane później tego dnia i pozostałe na placu wieczorem (do sprzedaży w kolejnych dniach).

88.3.

Którego dnia firma po raz pierwszy sprzeda mniej choinek, niż wynikałoby to z zainteresowania klientów (mniej niż L)?

88.4.

Zaproponuj przykładowy harmonogram dostaw po 50 drzewek jednego dnia, zgodnie z którym firma sprzeda możliwie najwięcej choinek i jednocześnie ostatniego dnia na placu pozostanie mniej niż 50 choinek. Podaj liczbę dostaw, dni tych dostaw oraz liczbę choinek sprzedanych w całym okresie sprzedaży.

88.5.

Podaj maksymalną liczbę choinek, jaka powinna być przy opisanym popycie codziennie (czyli 24 razy) dostarczana, aby 24 grudnia wieczorem na placu zostało mniej choinek niż było ich przywiezionych. Podaj liczbę choinek, które pozostaną na placu 24 grudnia wieczorem.

2.3. Przetwarzanie i tworzenie informacji

Zadanie 89.

Wiązka zadań *Punkty rekrutacyjne*

W postępowaniu rekrutacyjnym przy przyjmowaniu do pierwszej klasy szkoły ponadgimnazjalnej bierze się pod uwagę wyniki egzaminu gimnazjalnego, oceny ze świadectwa ukończenia gimnazjum oraz dodatkowe osiągnięcia ucznia. Maksymalnie można uzyskać 100 punktów, w tym:

1) 50 punktów za wyniki egzaminu gimnazjalnego, które oblicza się zgodnie z poniższymi zasadami:
   a) liczba punktów rekrutacyjnych za każdy wynik procentowy z zakresu: języka polskiego, historii i wiedzy o społeczeństwie, matematyki, przedmiotów przyrodniczych, języka obcego jest równa liczbie określającej wynik procentowy, uzyskany z danego zakresu, podzielonej przez dziesięć.
   b) liczba punktów rekrutacyjnych za wyniki egzaminu gimnazjalnego jest równa sumie punktów rekrutacyjnych uzyskanych z wymienionych wyżej zakresów egzaminu.

      Przykład:

      Uczeń uzyskał następujące wyniki z egzaminu gimnazjalnego z poszczególnych zakresów:

      — język polski — 74%
      — historia i wiedza o społeczeństwie — 88%
      — matematyka — 60%
      — przedmioty przyrodnicze — 71%
      — język obcy — 79%

      Liczba punktów rekrutacyjnych wynosi: 74 : 10 + 88 : 10 + 60 : 10 + 71 : 10 + 79 : 10 = 7,4 + 8,8 + 6 +7,1 + 7,9 = 37,2
2) 40 punktów za oceny z języka polskiego, matematyki, biologii i geografii otrzymane na świadectwie ukończenia gimnazjum, przyznawanych zgodnie z przeliczeniem przedstawionym w tabeli:

liczba punktów	ocena
0	dopuszczająca
4	dostateczna
6	dobra
8	bardzo dobra
10	celująca

3) 10 punktów za dodatkowe osiągnięcia, w tym:
   a) 2 punkty za ukończenie gimnazjum ze wzorową (6) oceną z zachowania,
   b) maksymalnie 8 punktów za szczególne osiągnięcia ucznia wymienione na świadectwie ukończenia gimnazjum.

W pliku `punkty_rekrutacyjne.txt` znajduje się 514 wierszy z danymi uczniów, potrzebnymi do przeprowadzenia rekrutacji. Pierwszy wiersz pliku jest wierszem nagłówkowym. Kolejne wiersze składają się z następujących informacji: nazwisko ucznia (`Nazwisko`), imię ucznia (`Imie`), liczba punktów za osiągnięcia (`Osiagniecia`), ocena z zachowania (`Zachowanie`), oceny na świadectwie ucznia z czterech przedmiotów branych pod uwagę przy rekrutacji: języka polskiego (`JP`), matematyki (`Mat`), biologii (`Biol`) i geografii (`Geog`), oraz wyniki egzaminów gimnazjalnych z zakresów: języka polskiego (`GHP`), historii i wiedzy o społeczeństwie (`GHH`), matematyki (`GMM`), przedmiotów przyrodniczych (`GMP`), języka obcego (`GJP`). Wszystkie dane liczbowe są prezentowane w postaci liczb całkowitych nieujemnych, przy czym oceny z zachowania i z przedmiotów mieszczą się w przedziale <2, 6>, wyniki za osiągnięcia w przedziale <0, 10>, zaś wyniki procentowe egzaminów gimnazjalnych w przedziale <0, 100>. Dane w wierszach pliku rozdzielone są średnikami.

Przykład fragmentu danych

```
Nazwisko;Imie;Osiagniecia;Zachowanie;JP;Mat;Biol;Geog;GHP;GHH;GMM;GMP;GJP
Swistek;Damian;0;4;4;5;6;6;62;13;26;67;62
Kowalik;Mateusz;7;4;4;2;5;6;90;8;21;52;33
```

Przykład obliczenia liczby punktów rekrutacyjnych

Swistek Damian – liczba punktów rekrutacyjnych = 0+0+(6+8+10+10)+(62+13+26+67+62)/10=57
Kowalik Mateusz – liczba punktów rekrutacyjnych = 7+0+(6+0+8+10)+ (90+8+21+52+33)/10=51,4

Korzystając z dostępnych narzędzi informatycznych, rozwiąż poniższe zadania. Odpowiedzi zapisz do pliku `wyniki_punkty.txt` (z wyjątkiem wykresu do zadania 4), a każdą odpowiedź poprzedź cyfrą oznaczającą to zadanie.

89.1.

Utwórz zestawienie uczniów, którzy spełniają jednocześnie następujące warunki: mają 0 punktów za osiągnięcia, co najmniej bardzo dobrą ocenę (5) z zachowania oraz średnią z przedmiotów punktowanych w rekrutacji większą od 4. Zestawienie uporządkuj alfabetycznie według nazwisk. Podaj imiona i nazwiska pierwszych 5 osób.

89.2.

Oblicz zgodnie z zasadami opisanymi w treści zadania, ile punktów rekrutacyjnych uzyskał każdy z uczniów. Podaj liczbę punktów rekrutacyjnych występującą najczęściej oraz listę nazwisk i imion uczniów, którzy uzyskali tę liczbę punktów.

89.3.

Utwórz zestawienie zawierające imiona i nazwiska uczniów, którzy uzyskali 100% punktów z co najmniej 3 zakresów egzaminu gimnazjalnego.

89.4.

Utwórz zestawienie zawierające rozkład ocen dla każdego punktowanego w rekrutacji przedmiotu (przykład schematu zestawienia poniżej). Dla otrzymanego zestawienia wykonaj wykres procentowy skumulowany. Zadbaj o czytelny opis wykresu.

Liczba ocen	**Język polski**	**Matematyka**	**Biologia**	**Geografia**
dopuszczających				
dostatecznych				
dobrych				
bardzo dobrych				
celujących				

89.5.

Podaj liczbę uczniów, którzy uzyskali łącznie więcej punktów rekrutacyjnych za oceny z przedmiotów punktowanych i dodatkowe osiągnięcia (w tym ocenę z zachowania) niż za wyniki egzaminu gimnazjalnego.

Zadanie 90.

Wiązka zadań *Akademiki*

W pliku `studenci.txt` zamieszczono listę 1616 studentów z 10 różnych uczelni, którzy starają się o miejsce w akademikach miasteczka studenckiego Mordor. W osobnych wier-

szach zostały zapisane informacje o każdym studencie: identyfikator studenta (`Id_studenta`), imię (`Imie`), nazwisko (`Nazwisko`), nazwa uczelni, na której on studiuje (`Uczelnia`), miejsce zamieszkania (`Miejsce_zam`), rok studiów (`Rok_studiow`) oraz dochód przypadający na jedną osobę w rodzinie wyrażony w denarach (`Dochod_na_osobe`). Pierwszy wiersz pliku jest wierszem nagłówkowym, a dane w wierszu są rozdzielone znakiem średnika.

Przykład

```
Id_studenta;Imie;Nazwisko;Uczelnia;Miejsce_zam;Rok_studiow;Dochod_na_osobe
1;Jadwiga;Baranowska;Uniwersytet Krolewski;Krosno;I;2382
2;Zofia;Chorzowska;Akademia Sztuk Pieknych;Pulawy;II;1549
3;Katarzyna;Bilska;Uniwersytet Krolewski;Lubliniec;I;3037
```

Korzystając z danych umieszczonych w pliku `studenci.txt`, wykonaj podane niżej zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku `wyniki.txt` (poza wykresem do zadania 88.4), a każdą z nich poprzedź numerem zadania.

90.1.

Podaj nazwę miejscowości (miejsce zamieszkania), z której pochodzi najwięcej studentów starających się o akademiki, oraz liczbę studentów z tej miejscowości. Ponadto podaj nazwę uczelni, na której studiuje najwięcej studentów z tej miejscowości.

90.2.

W akademikach jest w sumie 1000 miejsc. Początkowo Rada Mordoru ustaliła, że miejsce w akademiku otrzymają studenci, u których w rodzinach dochód na jedną osobę nie przekracza 2000 denarów. Podaj, ile miejsc pozostałoby wolnych, gdyby obowiązywało takie kryterium przydziału miejsc w akademikach.

Rada Mordoru postanowiła obsadzić wszystkie miejsca w akademikach. Podaj, jak należy zmienić próg kryterium dochodowego, aby dokładnie 1000 miejsc zostało wykorzystanych przez studentów. Próg musi być liczbą całkowitą. Taki próg istnieje dla podanych danych.

90.3.

Wykonaj zestawienie zawierające nazwę uczelni oraz średni dochód na osobę w rodzinie studentów tej uczelni, którzy starają się o pokój w Mordorze. Średni dochód podaj z dokładnością do dwóch miejsc po przecinku.

Do wykonanego zestawienia utwórz wykres kolumnowy. Aby uwypuklić różnicę w średnim dochodzie na osobę, przeskaluj oś OY tak, aby w początku układu współrzędnych znajdowała się wartość 1800 denarów.

90.4.

Dla każdej uczelni podaj liczbę, starających się o miejsce w Mordorze, studentów studiujących na I, II, III, IV i V roku studiów.

90.5.

Wśród studentów są rodzeństwa. Rozpoznać je można po wspólnym nazwisku (z dokładnością do ostatniej litery, np. Nowicka i Nowicki mogą być rodzeństwem), takim samym miejscu zamieszkania i takim samym dochodzie na osobę. Wypisz wszystkie rodzeństwa wśród

podanych studentów. W zestawieniu podaj: imię, nazwisko, miejsce zamieszkania i dochód na osobę każdego rodzeństwa.

Zadanie 91.

Wiązka zadań *Numery PESEL*

Numer PESEL to 11-cyfrowy kod jednoznacznie identyfikujący określoną osobę fizyczną. Dla wszystkich urodzonych w latach 1900-1999 skonstruowany został w następujący sposób:

- cyfry na pozycjach od 1 do 6 to data urodzenia (w kolejności: ostatnie dwie cyfry roku, numer miesiąca w postaci dwóch cyfr oraz numer dnia w postaci dwóch cyfr),
- cyfry na pozycjach od 7 do 9 to liczba porządkowa,
- cyfra na pozycji 10 oznacza płeć (cyfra parzysta dla kobiet, nieparzysta dla mężczyzn),
- cyfra z pozycji 11 to cyfra kontrolna.

Dla osób urodzonych w roku 2000 oraz w późniejszych latach do numeru miesiąca dodana jest liczba 20. W ten sposób w numerze PESEL odróżniane są od siebie dwa stulecia.

W pliku `pesele.txt` znajdują się 494 wiersze z danymi osób zarejestrowanymi w systemie ewidencji ludności. Pierwszy wiersz pliku jest wierszem nagłówkowym. Kolejne wiersze składają się z następujących informacji: numer PESEL (`PESEL`), nazwisko (`Nazwisko`) oraz imię (`Imie`).[2] Dane w wierszach pliku rozdzielone są średnikami.

Przykład

```
PESEL;Nazwisko;Imie
08242501475;Micun;Krzysztof
```

Korzystając z dostępnych narzędzi informatycznych, podaj odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku `pesele_wyniki.txt` (z wyjątkiem wykresu do zadania 4), a każdą odpowiedź poprzedź cyfrą oznaczającą to zadanie.

91.1.

Wypisz imiona kobiet występujące w pliku, które **nie** kończą się na literę „a”. Aby ustalić, które wiersze w pliku zawierają dane kobiet, skorzystaj z cyfry numeru PESEL oznaczającej płeć osoby.

91.2.

Wśród osób, których dane zostały zapisane w pliku, są takie, które mają to samo imię i nazwisko. Wskaż te osoby, podając ich numery PESEL, imiona i nazwiska.

91.3.

Wypisz imiona i nazwiska osób, których liczba porządkowa w numerze PESEL (zapisana na pozycjach 7-9) jest odpowiednio największa i najmniejsza.

91.4.

Utwórz zestawienie zawierające nazwy wszystkich miesięcy roku oraz liczbę osób urodzonych w poszczególnych miesiącach. Dla otrzymanego zestawienia wykonaj wykres kolumnowy. Zadbaj o czytelny opis wykresu.

---

[2] Z uwagi na prawną ochronę danych osobowych, imiona i nazwiska w podanym pliku są fikcyjne.

91.5.

Dla każdej osoby utwórz jej identyfikator złożony z pierwszej litery imienia, trzech pierwszych liter nazwiska oraz ostatniej cyfry numeru PESEL. Podaj w porządku alfabetycznym identyfikatory, które wystąpią więcej niż jeden raz.

Zadanie 92.

Wiązka zadań *Olimpiady*

W pliku o nazwie `dane_medale.txt` znajdują się informacje o liczbie medali zdobytych przez poszczególne państwa uczestniczące w letnich i zimowych igrzyskach olimpijskich w latach 1896 – 2014. W każdym wierszu znajdują się następujące informacje, oddzielone pojedynczymi znakami tabulacji: nazwa państwa (`Panstwo`), kontynent (`Kontynent`), liczba olimpiad letnich, w których dane państwo brało udział (`OL_letnie`), liczba poszczególnych medali zdobytych w olimpiadach letnich (`Zloty; Srebrny; Brazowy`), liczba olimpiad zimowych, w których brało udział państwo (`OL_zimowe`), liczba poszczególnych medali zdobytych w olimpiadach zimowych (`Zloty; Srebrny; Brazowy`). Pierwszy wiersz jest wierszem nagłówkowym.

Przykład

```
Panstwo    Kontynent OL_letnie  Zloty  Srebrny Brazowy OL_zimowe Zloty  Srebrny Brazowy
Afganistan Azja      13         0      0       2       0         0      0       0
Algieria   Afryka    12         5      2       8       3         0      0       0
```

Dane opracowane na podstawie:
http://pl.wikipedia.org/wiki/Klasyfikacja_medalowa_wszech_czas%C3%B3w_igrzysk_olimpijskich#cite_note-1

Wykorzystując dane zawarte w tym pliku i dostępne narzędzia informatyczne, wykonaj poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_olimpiady.txt` (z wyjątkiem wykresu do zadania 2). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 2 umieść w pliku `wykres_olimpiady.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku.

92.1.

Podaj liczbę państw, z których każde spełnia poniższe warunki:

- brało udział w co najmniej jednej olimpiadzie letniej,
- brało udział w co najmniej jednej olimpiadzie zimowej,
- zdobyło co najmniej 1 medal na olimpiadadch letnich, a nie zdobyło **żadnego medalu na olimpiadach zimowych**.

Podaj łączną liczbę medali zdobytych na olimpiadach letnich przez państwa spełniające podane warunki.

92.2.

Przyjmijmy, że każdemu kontynentowi przydzielamy za daną olimpiadę tyle punktów, ile państw z tego kontynentu w niej wystąpiło. Utwórz zestawienie zawierające dla każdego kontynentu łączną liczbę punktów zz olimpiady letnie oraz łączną liczbę punktów za olimpiady zimowe. Dla otrzymanego zestawienia sporządź wykres procentowy, skumulowany słupkowy. Pamiętaj o prawidłowym i czytelnym opisie wykresu.

92.3.

Podaj nazwy państw, który zdobyły na wszystkich olimpiadach (letnich i zimowych) więcej medali złotych niż (łącznie) medali srebrnych i brązowych.

92.4.

Dla każdego kontynentu podaj nazwę państwa z tego kontynentu, które zdobyło łącznie największą liczbę medali na wszystkich olimpiadach, oraz liczbę tych medali.

92.5.

Kraj nazywamy *letnim*, jeżeli **dla każdego typu medali** (złoty, srebrny, brązowy) kraj ten zdobył **więcej** medali tego typu w olimpiadach letnich niż w olimpiadach zimowych. Podobnie kraj nazywamy *zimowym*, jeżeli dla każdego typu medali zdobył on więcej medali tego typu w olimpiadach zimowych niż w olimpiadach letnich. Podaj, ile jest w Europie krajów letnich i ile jest krajów zimowych.

Zadanie 93.

Wiązka zadań *Podróże*

Warszawska firma komputerowa *Sofcik* ma swoich klientów w 13 miastach i wysyła do nich swoich 101 pracowników. W pliku ***podróże.txt*** zapisane zostały wszystkie wyjazdy pracowników firmy do klientów w 2014 roku.

Pierwszy wiersz pliku jest wierszem nagłówkowym. W każdym wierszu zapisano: imię (`Imie`) i nazwisko pracownika (`Nazwisko`), miasto, do którego pracownik był delegowany (`Miasto`), datę wyjazdu (`D_wyj`) i datę powrotu z delegacji (`D_powr`) oraz koszt wyjazdu (`Koszt_wyj`). Daty w pliku są podawane w formacie: `RRRR-MM-DD`.

Dane w wierszach pliku rozdzielone są pojedynczymi znakami tabulacji.

Przykład

```
Imie     Nazwisko  Miasto    D_wyj        D_powr      Koszt_wyj
Karolina Arska     Malbork   2014-01-02   2014-01-03  891,00
Justyna  Kolska    Siedlce   2014-01-02   2014-01-03  295,40
Dorota   Morska    Radom     2014-01-02   2014-01-03  302,50
```

Wykorzystując dane zawarte w tym pliku i dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_podroze.txt` (z wyjątkiem wykresu do zadania 4). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 4 umieść w pliku `wykres_podroze.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku.

93.1.

Podaj imię i nazwisko osoby, która w ciągu całego roku była tylko raz w delegacji.

93.2.

Utwórz zestawienie zawierające nazwiska osób, które w sumie były w delegacji więcej niż 40 dni. Dla każdej osoby podaj liczbę dni, jaką spędziła ona w delegacji.

93.3.

Na koszt delegacji składa się koszt wyjazdu oraz koszt diety za każdy dzień podróży.

Dieta obliczana jest zgodnie z następującą zasadą: jeżeli pracownikowi nie zapewniono posiłku, to dieta wynosi 30 zł za każdy dzień, jeżeli zaś pracownik jadł w hotelu śniadanie (co zdarza się zawsze po noclegu w hotelu), to dieta wynosi 24 zł.

Przykład

Pracownik był w delegacji 4 dni, koszt jego diety wynosi 30 zł (za pierwszy dzień)+ 3 · 24 zł (za 3 dni ze śniadaniem w hotelu), razem 102 zł.

Utwórz zestawienie zawierające informacje o sumie kosztów (koszt wyjazdów), jakie firma poniosła na delegacje pracowników do każdego z 13 miast. Zestawienie posortuj rosnąco ze względu na kwotę kosztów.

93.4.

Utwórz zestawienie zawierające łączną liczbę wyjazdów pracowników w delegacje w poszczególnych miesiącach. Sporządź wykres kolumnowy przedstawiający otrzymane zestawienie. Pamiętaj o prawidłowym i czytelnym opisie wykresu.

Uwaga: dla każdego miesiąca weź pod uwagę tylko te wyjazdy w delegacje, których data wyjazdu (`D_wyj`) znajduje się w tym miesiącu.

93.5.

Podaj średnią liczbę noclegów pracowników we **wszystkich** delegacjach oraz średnią liczbę noclegów pracowników przebywających w delegacjach, które były **co najmniej 2-dniowe**. Wyniki podaj w zaokrągleniu do dwóch miejsc po przecinku.

Zadanie 94.

Wiązka zadań *Centyle*

W celu uaktualnienia siatek centylowych wzrostu dzieci wykonywano regularne pomiary wzrostu 2 262 wylosowanych dzieci. W pliku `wzrost.txt` zapisano w kolejnych kolumnach, rozdzielonych średnikiem, identyfikator dziecka, jego płeć, długość ciała dziecka w dniu narodzin, a następnie wzrost po ukończeniu każdego kolejnego roku życia. Długość i wzrost mierzymy w centymetrach. Pierwszy wiersz jest wierszem nagłówkowym.

Przykład

```
ID_dz;plec;dl_ur;1rok;2lata;3lata;4lata;5lat;6lat;7lat;8lat;9lat;10lat;11lat;12lat;
13lat;14lat;15lat;16lat;17lat;18lat;19lat
1;d;54;72;88;97;105;112;118;124;130;136;142;149;155;161;164;166;167;167;168;168
2;d;46;64;82;91;99;106;111;117;123;128;134;141;146;151;154;156;156;156;157;157
3;ch;54;75;88;97;104;111;117;123;129;134;139;145;151;158;165;171;175;177;178;179
```

Wykorzystując dane zawarte w pliku `wzrost.txt` i dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w jednym pliku o nazwie `wyniki` (z wyjątkiem wykresu do zadania 6). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 6 umieść w pliku `wykres.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku z Twoim rozwiązaniem.

94.1.

Znajdź dziecko, które od urodzenia do ukończenia 19 roku życia osiągnęło największy przyrost wysokości ciała, wyrażony procentowo w stosunku do długości ciała w dniu urodzin. Podaj ten przyrost oraz identyfikator dziecka, które uzyskało ten przyrost.

94.2.

Podaj , w którym roku życia (od 0 do 19 lat) średnia arytmetyczna wzrostu chłopców jest mniejsza o co najmniej 1 cm od średniej arytmetycznej wzrostu dziewcząt. Dzieci w dniu urodzin są w wieku 0.

94.3.

Podaj liczbę dzieci, które przestały rosnąć po ukończeniu 15 lat.

94.4.

Przyrost w n-tym roku życia to różnica między wzrostem po ukończeniu n lat a wzrostem po ukończeniu $n - 1$ lat.

Przyrost wysokości ciała z każdym kolejnym rokiem życia dziecka maleje, jednak tuż przed okresem dojrzewania zmienia trend i zaczyna się zwiększać. Oblicz średni przyrost wysokości ciała w kolejnych latach, dla każdej płci osobno. Na tej podstawie podaj, od którego roku życia dziewcząt i od którego roku życia chłopców przyrost wysokości ciała zaczyna się zwiększać.

94.5.

Centylem rzędu p (p-tym centylem) wzrostu dzieci nazywamy taki najmniejszy wzrost, że co najmniej p% dzieci ma wzrost mniejszy lub równy tej liczbie oraz co najmniej 100% – p% dzieci ma wzrost większy lub równy tej liczbie, przy czym liczba p jest liczbą całkowitą i $p \in (0; 100\%)$.

Na przykład 95. centyl wzrostu chłopców to taki wzrost, że co najmniej 95% chłopców ma wzrost mniejszy lub równy niż podana wartość, a co najmniej 5% ma wzrost większy lub równy podanej wartości.

Jeżeli liczba dzieci jest parzysta i granica tych dwóch grup wypada pomiędzy dwiema osobami o różnym wzroście należy podać średnią tych dwóch wartości.

Dla chłopców w wieku 1 roku, 10 lat i 19 lat oblicz 5. centyl oraz 95. centyl.

94.6.

Dla każdego badanego wieku chłopców (od 0 do 19) podaj medianę wzrostu.

Dla wykonanego zestawienia wykonaj wykres, którego pionowa oś będzie oznaczać wzrost, a pozioma — wiek. Pamiętaj o czytelnym opisie wykresu.

Zadanie 95.

Wiązka zadań *Giełda Papierów Wartościowych*

W pliku `gpw.txt` znajdują się dane dotyczące notowań akcji na Giełdzie Papierów Wartościowych w Warszawie w dniach: 21, 22 i 23 stycznia 2015.

W pierwszym wierszu pliku umieszczono nagłówki kolumn. Każdy następny wiersz pliku zawiera 7 danych dotyczących jednej spółki notowanych w jednej sesji: datę notowań (`data`), nazwę spółki (`nazwa`), jej identyfikator (`ISIN`), kurs jednej akcji na koniec dnia wyrażony w zł (`kurs_zamkniecia`), liczbę akcji spółki, które zmieniły właściciela w danym

dniu (`wolumen`), łączną wartość transakcji wszystkich akcji w danym dniu wyrażoną w zł (`obrot`) oraz liczbę akcji przyjętą do obliczenia indeksu WIG (`pakiet_wig`). Jeśli spółka nie wchodzi w skład portfela WIG, jej pole `pakiet_wig` ma wartość 0. Dane w wierszach oddzielone są znakami tabulacji.

Przykład

data	*nazwa*	*ISIN*	*kurs_zamkniecia*	*wolumen*	*obrot*	*pakiet_wig*
2015-01-21	ALTA	PLTRNSU00013	2,1	4664	9710	7353000
2015-01-22	ELZAB	PLELZAB00010	15,3	16599	249530	2716000
2015-01-23	PLAZA	NL0000686772	0,19	101576	19300	0

Dane pochodzą ze strony http://www.gpw.pl

Rozwiąż poniższe zadania, wykorzystując dostępne narzędzia informatyczne. Wyniki umieść w pliku tekstowym o nazwie `gpw_wyniki.txt`. Do oceny oddaj plik tekstowy zawierający wyniki oraz plik zawierający realizację komputerową rozwiązania.

95.1

Średni kurs akcji spółki w danym dniu obliczany jest jako iloraz *obrot*/*wolumen*. W przypadku gdy obrót i wolumen są równe 0 (czyli żadna akcja tej spółki nie zmieniła właściciela w ciągu dnia), średni kurs akcji spółki jest równy kursowi zamknięcia sesji.

Podaj **nazwy** trzech spółek, które w dniu **2015-01-21** osiągnęły najwyższe średnie kursy akcji, podaj również wartości tych kursów. Zapisz wyniki z dokładnością do 2 cyfr po przecinku.

95.2.

Dzienna zmiana procentowa kursu spółki jest obliczana na podstawie kursów zamknięcia z dwóch kolejnych dni:

$$d = \left(\frac{kurs\ zamknięcia\ z\ dnia\ bieżącego}{kurs\ zamknięcia\ z\ dnia\ poprzedniego} - 1\right) * 100\%.$$

Podaj nazwę spółki, która w dniu **2015-01-23** uzyskała największą dzienną zmianę kursu. Podaj wielkość tej zmiany w procentach, w zaokrągleniu do dwóch cyfr po przecinku.

95.3.

ISIN (*International Securities Identification Number*) jest międzynarodowym identyfikatorem spółki na rynku finansowym. Pierwsze dwa znaki stanowią kod kraju. Dla Polski przyjęto ”PL”. Wykorzystując tę informację:

a) podaj **liczbę spółek** krajowych i liczbę spółek zagranicznych notowanych na giełdzie w podanym okresie;

b) podaj **łączną wartość obrotów** spółek krajowych i łączną wartość obrotów spółek zagranicznych w ciągu trzech danych dni oraz **procentowy udział spółek krajowych** w łącznych obrotach wszystkich spółek. Wyniki podaj w zaokrągleniu do dwóch cyfr po przecinku.

95.4.

Warszawski Indeks Giełdowy WIG jest miarą koniunktury na giełdzie i wyraża łączną wartość spółek obecnych na Giełdzie Papierów Wartościowych (GPW) w stosunku do ich wartości w pierwszym dniu notowania (16 kwietnia 1991).

WIG obejmuje tylko wybrane spółki, spełniające kryteria co do procentu i wartości akcji w wolnym obrocie. Portfel spółek uczestniczących w WIG jest aktualizowany co kwartał. Obecnie (styczeń 2015) w portfelu jest 377 spółek. Dla każdej z nich określono *pakiet*, czyli liczbę akcji branych pod uwagę przy obliczaniu WIG. Wartość WIG oblicza się według wzoru:

$$WIG = \frac{M}{M_b * K} * W_b$$

gdzie: M — aktualna łączna wartość rynkowa pakietów wszystkich spółek w portfelu WIG, równa sumie iloczynów *pakiet*kurs* wszystkich spółek

M_b = 57 140 000 zł — wartość bazowa spółek na początku notowania;

W_b = 1000 — wartość bazowa indeksu WIG na początku notowania;

K = 96,482 137 39 — korektor wynikający ze zmian składu spółek zaliczanych do WIG.

WIG zmienia się wraz ze zmianą kursu akcji. Opierając się na wartościach kursów zamknięcia sesji, podaj dla każdego z trzech danych dni:

- łączną wartość rynkową pakietów wszystkich spółek przy kursie zamknięcia,
- wartość indeksu WIG zaokrągloną do dwóch cyfr po przecinku.

95.5.

Inwestor giełdowy śledzi notowania kursu zamknięcia sesji. Na podstawie wyników z trzech kolejnych dni następująco ocenia, co warto zrobić:

- jeżeli kurs spółki rośnie, i to rośnie coraz szybciej (drugi wzrost jest większy od pierwszego), warto kupić akcje,
- jeżeli kurs spółki spada, i to spada coraz szybciej (drugi spadek jest większy od pierwszego), warto sprzedać akcje,
- w pozostałych sytuacjach warto poczekać i obserwować rozwój sytuacji.

Wykorzystując notowania z trzech danych dni, podaj **liczbę spółek**, których akcje warto zdaniem inwestora **kupić**, liczbę spółek, których akcje jego zdaniem warto **sprzedać,** i liczbę spółek, które według niego warto dalej obserwować.

Zadanie 96.

Wiązka zadań *Prognoza liczby ludności w Polsce do roku 2050*

W pliku `ludnosc.txt` znajdują się dane: prognostyczne dotyczące liczby ludności Polski w latach 2013–2050 z podziałem na wiek (od 0 do 100 lat), płeć oraz miejsce zamieszkania: miasto lub wieś.

Pierwszy wiersz pliku jest wierszem nagłówkowym, a każdy następny wiersz pliku zawiera 6 liczb: rok kalendarzowy (`rok`), wiek osób w latach (`wiek`), liczbę mężczyzn (`m_miasto`) i liczbę kobiet w tym wieku (`k_miasto`) zamieszkałych w miastach oraz liczbę mężczyzn

(`m_wies`) i liczbę kobiet w tym wieku (`k_wies`) zamieszkałych na wsi. Osoby w wieku ponad 100 lat zaliczono dla uproszczenia do grupy 100-latków. Dane w wierszu oddzielone są znakiem tabulacji.

Przykład

```
rok     wiek    m_miasto  k_miasto   m_wies   k_wies
2013    0       107301    101414     77659    73821
2013    1       116232    109575     83756    79140
2013    2       118851    111765     85217    80842
```

Dane pochodzą ze strony http://stat.gov.pl

Rozwiąż poniższe zadania, wykorzystując dostępne narzędzia informatyczne. Wyniki umieść w pliku tekstowym o nazwie `ludnosc_wyniki.txt`. Do oceny oddaj plik tekstowy zawierający wyniki oraz plik zawierający realizację komputerową rozwiązania.

96.1.

Porównaj na wspólnym wykresie strukturę wiekową ludności Polski w roku 2013 i 2050. Sporządź wykres typu *Punktowy*, zawierający dwie serie danych: dla roku 2013 i roku 2050, który będzie przedstawiał dla każdego wieku od 0 do 100 łączną liczbę ludności w tym wieku.

96.2.

Podaj **stosunek** liczby ludności miast do liczby ludności wsi obliczony dla roku 2013 i dla roku 2050. Wyniki zapisz w postaci zaokrąglonej do dwóch cyfr po przecinku.

96.3.

Podaj średni wiek **mężczyzny** zamieszkałego **w mieście** w roku 2013 oraz średni wiek **mężczyzny** zamieszkałego **w mieście** w roku 2050.

Wyniki zapisz w postaci zaokrąglonej do liczby całkowitej.

Uwaga: Dla danego roku należy obliczyć średnią ważoną po wszystkich grupach wiekowych od 0 do 100:

$$\textit{średni wiek} = \frac{0 * m_0 + 1 * m_1 + 2 * m_2 + \cdots + 99 * m_{99} + 100 * m_{100}}{m_0 + m_1 + \cdots + m_{99} + m_{100}}$$

gdzie m_i — liczba mężczyzn w mieście w grupie wiekowej *i* lat.

96.4.

Strukturę ludności charakteryzuje taka prawidłowość, że w każdym roku kalendarzowym w młodszych rocznikach mężczyźni stanowią większość, jednak wraz z wiekiem ta przewaga się zmniejsza. W pewnym wieku zaczynają przeważać liczebnie kobiety.

Sporządź zestawienie, w którym dla każdego roku kalendarzowego z zakresu 2013-2050 podasz najniższy wiek, w jakim kobiety przeważają liczebnie mężczyzn.

96.5.

Rozważ trzy grupy wiekowe:

- młodzież do 18 roku życia włącznie;
- osoby w wieku produkcyjnym od 19 do 67 roku życia włącznie;
- emeryci powyżej 67 roku życia.

Dla każdego roku kalendarzowego oblicz liczbę ludności w każdej z tych grup.

Sporządź wykres kolumnowy skumulowany procentowy, ilustrujący liczebność tych trzech grup w kolejnych latach okresu 2013–2050.

Sporządź zestawienie, w którym dla każdego roku podasz, jaki procent ludności stanowią osoby w wieku produkcyjnym. Wynik zapisz w postaci zaokrąglonej do całkowitej liczby procentów.

Zadanie 97.

Wiązka zadań *Stopa bezrobocia*

W pliku o nazwie `stopa_bezrobocia.txt` znajdują się miesięczne stopy bezrobocia w państwie Klingonów w latach 1945–2014. W każdym wierszu znajduje się 13 liczb, oddzielonych pojedynczymi znakami tabulacji: rok oraz stopy bezrobocia w kolejnych dwunastu miesiącach. Pierwszy wiersz jest wierszem nagłówkowym.

Przykład

ROK	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
1945	3,2	4,2	4,1	3,9	3,9	4,0	4,0	4,2	4,4	4,1	4,0	3,8
1946	4,5	3,9	3,6	3,6	3,7	3,5	3,5	3,6	3,7	3,9	4,1	4,5
1947	5,0	5,9	6,2	6,7	6,9	6,9	6,6	6,8	7,0	7,1	6,7	6,3

Wykorzystując dane zawarte w tym pliku i dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_bezrobocie.txt` (z wyjątkiem wykresu do zadania 3). Odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie. Wykres do zadania 3 umieść w pliku `wyniki_bezrobocie.xxx`, gdzie `xxx` oznacza rozszerzenie odpowiednie dla formatu pliku.

97.1.

Miesiącem wysokiego zagrożenia dla gospodarki w państwie Klingonów nazywamy taki miesiąc, w którym stopa bezrobocia jest większa niż 10. Podaj liczbę miesięcy wysokiego zagrożenia, jakie wystąpiły latach 1945–2014.

97.2.

Podaj najniższą i najwyższą średnią roczną stopę bezrobocia (wyniki podaj w zaokrągleniu do dwóch miejsc po przecinku) oraz rok, w którym one wystąpiły.

97.3.

Utwórz zestawienie zawierające dla każdego roku w latach 1945–2014 minimalną i maksymalną miesięczną stopę bezrobocia odnotowaną w tych latach.

Dla otrzymanego zestawienia sporządź wykres liniowy. Pamiętaj o prawidłowym i czytelnym opisie wykresu.

97.4.

Znajdź najdłuższy nierosnący ciąg miesięcznych stóp bezrobocia w kolejnych miesiącach w latach 1945–2014. Podaj miesiąc i rok początkowy oraz miesiąc i rok końcowy znalezionego ciągu oraz jego długość.

Przykład

Dla pliku zawierającego następujące dane:

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII
2002	12,90	12,90	12,50	13,00	12,50	12,20	11,00	11,70	11,50	11,30	11,40	11,60
2003	14,20	14,40	14,30	**14,40**	**13,60**	**13,20**	**13,10**	**13,00**	**13,00**	**13,00**	**12,90**	**12,80**
2004	**12,50**	12,60	13,30	12,90	12,60	12,30	12,30	12,40	12,40	12,50	12,90	13,40

długość najdłuższego nierosnącego ciągu stóp bezrobocia wynosi 10 (ciąg liczb: 14,40; 13,60; 13,20; 13,10; 13,00; 13,00; 13,00; 12,90; 12,80; 12,50). Miesiąc i rok początku ciągu to IV 2003. Miesiąc i rok końca ciągu to I 2004.

97.5.

Podaj liczbę lat, w których stopa bezrobocia w każdym miesiącu była większa od stopy bezrobocia w tym samym miesiącu roku poprzedniego.

2.4. Bazy danych

Zadanie 98.

Wiązka zadań *Dziennik ocen*

Dane są trzy pliki tekstowe: `uczniowie.txt`, `przedmioty.txt` i `oceny.txt`, w których zapisano oceny wystawiane uczniom w pewnym technikum informatycznym w okresie od 1.09.2014 r. do 18.12.2014 r.

Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a kolumny w wierszach rozdzielone są znakami tabulacji.

Plik o nazwie `uczniowie.txt` zawiera informacje dotyczące uczniów szkoły. W każdym wierszu znajduje się: identyfikator ucznia (`ID_ucznia`), jego imię (`Imie`), nazwisko (`Nazwisko`) oraz oznaczenie klasy za pomocą rzymskiej liczby i litery (`Klasa`).

Przykład

```
Id_ucznia Imie      Nazwisko       Klasa
123/2011  Wojciech  Banasik        IV E
124/2011  Monika    Baranowska     IV E
125/2011  Janusz    Czerwinski     IV E
```

Plik `przedmioty.txt` zawiera identyfikator przedmiotu (`Id_przedmiotu`) oraz nazwę przedmiotu (`Nazwa_przedmiotu`).

Przykład

```
Id_przedmiotu  Nazwa_przedmiotu
1              j.polski
2              j.angielski
3              j.niemiecki
```

W pliku `oceny.txt` zapisane są w każdym wierszu: identyfikator oceny (`Id_oceny`), data wystawienia oceny (`Data`), identyfikator ucznia (`Id_ucznia`), identyfikator przedmiotu (`Id_przedmiotu`) oraz ocena (`Ocena`).

Przykład

```
Id_oceny  Data        Id_ucznia Id_przedmiotu  Ocena
1         2014-09-08  704/2014  1              2
2         2014-09-08  312/2012  1              4
3         2014-09-08  649/2013  3              5
```

Korzystając z danych zawartych w tych plikach oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź umieść w pliku `wyniki.txt`, poprzedzając ją numerem odpowiedniego zadania.

98.1.

Imiona dziewcząt w zestawieniu kończą się na literę „a”. Podaj klasy, w których ponad 50% wszystkich uczniów to dziewczęta.

98.2.

Podaj daty, kiedy w szkole wystawiono więcej niż 10 jedynek jednego dnia.

98.3.

Podaj, z dokładnością do dwóch miejsc po przecinku, średnie ocen z języka polskiego dla każdej klasy czwartej.

98.4.

Podaj zestawienie zawierające dla każdego przedmiotu liczbę piątek wystawionych w kolejnych miesiącach od września do grudnia łącznie we wszystkich klasach.

98.5.

Podaj zestawienie imion i nazwisk uczniów klasy II A, którzy nie otrzymali żadnej oceny z przedmiotu sieci komputerowe.

Zadanie 99.

Wiązka zadań *Bezpieczeństwo w szkole*

Wydział Edukacji powiatu Bajtolandia przeprowadził wśród uczniów szkół badanie na temat bezpieczeństwa w szkole. W ankiecie udział wzięli uczniowie ze 130 wybranych szkół różnego typu.

W plikach `ankiety.txt`, `szkoly.txt` i `gminy.txt` znajdują się dane dotyczące ankietowanych szkół w powiecie Bajtolandia oraz wyniki ankiet wypełnionych przez uczniów tych szkół. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

W kolejnych wierszach pliku `ankiety.txt` znajdują się wyniki 5600 ankiet: numer ankiety (`Nr_ankiety`), oceny podane w odpowiedzi na poszczególne pytania (`pyt1`, `pyt2`, ..., `pyt6`), `Plec` (*k* — dziewczyna, *m* — chłopak), identyfikator szkoły (`Id_szkoły`).

Przykład

Nr_ankiety	pyt1	pyt2	pyt3	pyt4	pyt5	pyt6	Id_szkoly	Plec
a0001	1	2	2	3	3	4	S001	k
a0002	5	3	3	2	2	3	S001	k
a0003	5	3	5	5	1	4	S001	m
a0004	3	2	2	1	2	3	S001	k

Treść pytań ankiety:

pyt1: Czy w swojej szkole czujesz się bezpiecznie?

pyt2: Czy byłeś na terenie szkoły ofiarą niebezpiecznych zachowań, agresji?

pyt3: Czy byłeś świadkiem niebezpiecznych sytuacji, agresji w szkole?

pyt4: Czy zdarzyło Ci się zrobić komukolwiek jakąś przykrość?

pyt5: Czy nauczyciele rozmawiają z uczniami o przemocy?

pyt6: Czy w szkole odbywają się zajęcia (spotkania/lekcje/wystawy/inne formy) na temat bezpieczeństwa w szkole?

Odpowiedzi na pytania to oceny w skali od 1 do 5, oznaczające odpowiednio:

1. nigdy
2. bardzo rzadko
3. rzadko
4. często
5. bardzo często

W pliku `szkoly.txt` znajduje się 130 wierszy z informacjami o szkołach: identyfikator szkoły (`Id_szkoly`), `Rodzaj_szkoly` (SP, G, LO, T, ZS), `Kod_gminy`.

Przykład

Id_szkoly	Rodzaj_szkoly	Kod_gminy
S001	LO	GM19
S002	SP	GM17
S003	LO	GM07
S004	T	GM19

W pliku `gminy.txt` każdy wiersz zawiera informacje o jednej z 20 gmin w Bajtolandii: `Kod_gminy`, `Nazwa_gminy`.

Przykład

```
Kod_gminy      Nazwa_gminy
GM01           Piatki Gorne
GM02           Piatki Dolne
GM03           Sobotka
```

Wykorzystując dane zawarte w tych plikach i dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie ***wyniki_ankiety.txt***. Odpowiedź do każdego zadania rozpocznij od nowego wiersza i poprzedź numerem oznaczającym to zadanie.

99.1.

Podaj liczbę wszystkich ankietowanych dziewcząt i wszystkich ankietowanych chłopców.

99.2.

Dla każdego rodzaju szkoły podaj średnią ocenę odpowiedzi na każde pytanie. Wyniki podaj w zaokrągleniu do dwóch miejsc po przecinku.

99.3.

Dla każdej gminy wyznacz średnią ocenę uczniów, z jej terenu podaną w odpowiedzi na ostatnie (szóste) pytanie. Wyniki umieść w zestawieniu zawierającym dwie kolumny: kod gminy, średnią ocenę uczniów. Zestawienie uporządkuj malejąco ze względu na średnią ocenę. Średnie podaj w zaokrągleniu do dwóch miejsc po przecinku.

99.4.

Utwórz zestawienie zawierające dla każdego rodzaju szkoły informacje o liczbie uczniów, którzy podali ocenę 5 na pytanie trzecie. Zestawienie posortuj alfabetycznie według rodzaju szkoły.

99.5.

Podaj nazwę gminy z największą liczbą uczniów biorących udział w badaniu oraz liczbę tych uczniów.

99.6.

Utwórz zestawienie zawierające informacje o liczbie dziewcząt i chłopców (osobno) z poszczególnych rodzajów szkół, którzy podali najwyższą ocenę 5 na pytanie 1.

Zadanie 100.

Wiązka zadań *E-learning*

Szkoła postanowiła wprowadzić dla uczniów dodatkowe zajęcia z informatyki z wykorzystaniem systemu e-learning. W plikach: `osoby.txt`, `listy.txt`, `punktacja.txt` znajdują się informacje na temat: uczniów szkoły, którzy uczyli się w systemie e-learning w okresie od 1.09.2014 do 15.02.2015, list zadań oraz uzyskanych przez uczniów wyników. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

Plik o nazwie `osoby.txt` zawiera 60 wierszy z informacjami na temat osób, które wysyłały rozwiązania zadań poprzez system. Są to: identyfikator osoby (`id_osoby`), jej nazwisko (`nazwisko`), imię (`imie`) oraz nazwa grupy, do której osoba została przydzielona (`grupa`).

Przykład

```
id_osoby  imie nazwisko  grupa
35   Joanna    Matura    G3
36   Anna      Piasecka  G3
37   Katarzyna Zienowicz G3
```

W pliku `listy.txt` znajduje się 11 wierszy z informacjami na temat list zadań zamieszczonych w systemie: numerem listy (`id_listy`), nazwą listy zadań (`nazwa`) oraz terminem oddania (`termin_oddania`).

Przykład

```
id_listy  nazwa     termin_oddania
7         C7        2015-01-26
8         P1        2014-11-10
```

Plik o nazwie `punktacja.txt` zawiera 653 wiersze z informacjami o wynikach uczniów. Są to: liczba porządkowa (`lp`), identyfikator danej osoby (`id_osoby`), identyfikator listy zadań (`id_listy`), liczba punktów zdobytych przez daną osobę (`punkty`) oraz data przesłania rozwiązania listy zadań (`data`).

Przykład

```
lp   id_osoby  id_listy  punkty    data
1    1         1         12        2014-10-20
2    1         2         12        2014-11-03
3    1         8         22        2014-11-10
```

Wykorzystując dane zawarte w tych plikach i dostępne narzędzia informatyczne, wykonaj poniższe zadania, a wyniki zapisz w pliku o nazwie `wyniki_elearning.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

100.1.

Utwórz zestawienie, w którym dla każdej listy zadań podasz średnią liczbę punktów otrzymanych za te zadania przez uczniów. W zestawieniu podaj nazwę listy oraz średnią liczbę punktów zaokrągloną do dwóch miejsc po przecinku.

100.2.

Utwórz zestawienie, w którym podasz imiona i nazwiska osób, które spóźniły się o 14 lub więcej dni z oddaniem dowolnej listy o nazwie zaczynającej się od litery „P”.

100.3.

Na podstawie liczby wszystkich zdobytych przez uczniów punktów wystawione zostały oceny według zasad przedstawionych w tabeli:

przedział punktów	**ocena**
(0, 72)	1
[72, 90)	2
[90, 126)	3
[126, 153)	4
[153, 180)	5

Podaj, ile osób otrzymało ocenę 1, 2, 3, 4, 5.

100.4.

Utwórz czytelne zestawienie tabelaryczne, w którym dla każdej grupy podasz, ile osób otrzymało liczbę punktów równą 10, 11, 12. W swoich obliczeniach weź pod uwagę wszystkie listy zadań.

100.5.

Podaj imiona i nazwiska osób, które nie wysłały przynajmniej jednej listy zadań. Zestawienie posortuj rosnąco ze względu na nazwiska osób.

Zadanie 101.

Wiązka zadań *Karta MaturaSport*

Władze miasta wprowadziły program *MaturaSport*, w ramach którego opłacają zajęcia sportowe dla uczniów ostatnich klas szkół ponadgimnazjalnych. Uczniowie korzystają z kart *MaturaSport*, które umożliwiają wstęp do różnych obiektów sportowych i uczestnictwo w prowadzonych tam zajęciach. W następujących plikach zgromadzono dane dotyczące wykorzystania kart programu *MaturaSport* w kwietniu 2014 roku. Dane w plikach są oddzielone średnikami, a pierwszy wiersz zawiera nagłówki kolumn.

Każdy wiersz w pliku `osoby.txt` zawiera informacje o jednym użytkowniku karty: jego identyfikator (`Id_uzytkownika`), nazwisko (`Nazwisko`), imię (`Imie`) i płeć (`Plec`). Płeć jest oznaczona literą „K” lub „M”.

Przykład

```
Id_uzytkownika;Nazwisko;Imie;Plec
1;Olszowka;Klara;K
2;Wieruszewski;Antoni;M
```

W pliku `zajecia.txt` zawarte są informacje o zajęciach prowadzonych w obiektach sportowych, biorących udział w programie. Każdy wiersz zawiera: identyfikator zajęć (`Id_zajec`), nazwę obiektu (`Obiekt`), rodzaj zajęć (`Zajecia`) i koszt (`Koszt`) jednego wejścia na zajęcia opłacany z programu *MaturaSport.*

Przykład

```
Id_zajec;Obiekt;Zajecia;Koszt
1;Redeco;Basen;9
```

Każdy wiersz pliku `wejscia.txt` zawiera informacje o pojedynczym wejściu użytkownika karty na wybrane zajęcia: numer porządkowy (`Lp`), identyfikator użytkownika (`Id_uzytkownika`), datę (`Data`) i identyfikator zajęć (`Id_zajec`).

Przykład

```
Lp ;Id_uzytkownika;Data;Id_zajec
1;1;2014-04-05;16
```

Wykorzystując dane zawarte w plikach `osoby.txt`, `zajecia.txt`, `wejscia.txt`, rozwiąż poniższe zadania. Odpowiedzi zapisz do pliku `wyniki_sport.txt`, a każdą z nich poprzedź cyfrą oznaczającą właściwe zadanie.

101.1.

Podaj, ile kobiet i ilu mężczyzn uczęszczało na zajęcia `Fitness TBC`. Zwróć uwagę, że niektóre osoby mogły być na tych zajęciach kilkakrotnie, a w zestawieniu powinny być uwzględnione tylko raz.

101.2.

Utwórz zestawienie, w którym dla każdego obiektu podasz, jaką łączną kwotę zapłacono za prowadzone w nim zajęcia.

101.3.

Podaj nazwiska i imiona osób, które w dniu 16 kwietnia 2014 r. uczestniczyły w więcej niż jednych zajęciach.

101.4.

Podaj rodzaj zajęć, w których uczestniczyło najwięcej osób. Podaj podaj liczbę tych osób i nazwę obiektu, w którym te zajęcia były prowadzone.

101.5.

Utwórz zestawienie, w którym dla każdego obiektu podasz, ile odnotowano w nim wejść na zajęcia. Zestawienie uporządkuj alfabetycznie według nazw obiektów.

Zadanie 102.

Wiązka zadań *Portal społecznościowy*

Użytkownicy pewnego niszowego portalu społecznościowego mogą zawierać ze sobą znajomości, a także dzielić się zdjęciami. W plikach tekstowych podane są dane pewnej grupy użytkowników portalu, lista par użytkowników będących znajomymi, a także lista zdjęć, które opublikowali użytkownicy. Dane w wierszach rozdzielone są znakiem tabulacji, a pierwszy wiersz każdego pliku jest wierszem nagłówkowym.

Plik `uzytkownicy.txt` opisuje użytkowników portalu: dla każdego podany jest kolejno: jego identyfikator w bazie (`ID_uzytkownika`), imię (`Imie`), nazwisko (`Nazwisko`), kraj, z którego on pochodzi (`Kraj`) oraz płeć (`Plec`, M — mężczyzna, K — kobieta):

```
ID_uzytkownika Imie           Nazwisko  Kraj
     Plec
1              Joshua         King      Stany Zjednoczone   M
2              Leonardo       De Luca   Wlochy              M
3              Aleksandra     Krawczyk  Polska              K
```

Plik `znajomosci.txt` zawiera listę par znajomych. Każdy wiersz zawiera dwa różne identyfikatory użytkowników, którzy są znajomymi (`Znajomy_1` oraz `Znajomy_2`), oraz datę zawarcia znajomości (`Data`).

```
Znajomy_1 Znajomy_2 Data
64        12        2014-11-23
81        46        2013-01-27
82        13        2013-11-01
```

W pliku `zdjecia.txt` opisane są zdjęcia opublikowane przez użytkowników. Każdy wiersz zawiera: identyfikator zdjęcia (`ID_zdjecia`), datę publikacji zdjęcia (`Data_dodania`), identyfikator użytkownika, który je dodał (`ID_uzytkownika`), oraz podaną w pikselach szerokość i wysokość zdjęcia (`Szerokosc` i `Wysokosc`).

```
ID_zdjecia     Data_dodania   ID_uzytkownika Szerokosc Wysokosc
1              2013-06-29     11             720       480
2              2014-09-08     56             720       480
3              2013-08-30     74             1440      576
```

Wykorzystując dane zawarte w podanych plikach oraz dostępne narzędzia informatyczne, wykonaj poniższe polecenia. Każdą odpowiedź umieść w pliku `wyniki.txt`, poprzedzając ją numerem odpowiedniego zadania.

102.1.

Podaj, ile zdjęć zostało opublikowanych w 2014 roku.

102.2.

Podaj wszystkie pary użytkowników, którzy są znajomymi i mają takie samo imię. Dla każdej takiej pary podaj imię, nazwisko i kraj pochodzenia każdego użytkownika.

102.3.

Podaj zestawienie 10 krajów, z których pochodzi najwięcej zdjęć. Dla każdego kraju podaj jego nazwę i liczbę zdjęć z niego przesłanych. Posortuj listę malejąco według liczby opublikowanych zdjęć.

102.4.

Podaj wysokość i szerokość zdjęcia, które ma najwięcej pikseli, oraz imię i nazwisko użytkownika, które je opublikował. Możesz założyć, że jest tylko jedno takie zdjęcie.

102.5.

Znajdź wśród użytkowników mężczyzn, którzy opublikowali w portalu zdjęcie, nie mając w momencie jego publikacji żadnych znajomych. Dla każdego takiego użytkownika podaj jego imię, nazwisko i kraj pochodzenia. Listę posortuj rosnąco według alfabetycznej kolejności nazwisk użytkowników.

Zadanie 103.

Wiązka zadań *Rentgenodiagnostyka*

W okresie sezonu narciarskiego w pracowni rentgenodiagnostyki szpitalnego oddziału ratunkowego (SOR) wykonano 2361 zdjęć RTG.

W plikach `pacjenci.txt`, `badania.txt` i `rtg.txt` znajdują się informacje na temat pacjentów, wykonanych im zdjęć RTG oraz rodzajach badań RTG wykonywanych przez pracownię. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są średnikami.

W pliku `pacjenci.txt` znajdują się wiersze z informacjami o 816 pacjentach urodzonych przed rokiem 2000: `PESEL`, `Nazwisko`, `Imie`, `Plec` (*k* — kobieta, *m* — mężczyzna), `NFZ` (oddział Narodowego Funduszu Zdrowotnego, do którego należy pacjent).

Przykład

```
PESEL;Nazwisko;Imie;Plec;NFZ
37112515913;Latacki;Stanislaw;m;Opolski
38012109293;Stachniuk;Przemyslaw;m;Dolnoslaski
47101603441;Dubiel;Zdzislawa;k;Slaski
```

W pliku `badania.txt` znajdują się wiersze z informacjami o zdjęciach RTG wykonanych pacjentom: `PESEL`, `Id_badania`.

Przykład

```
PESEL;Id_badanie;
37112515913;Id_13;
38012109293;Id_16;
45032403378;Id_23;
```

Każdy wiersz pliku `rtg.txt` zawiera następujące informacje o wykonywanych w pracowni zdjęciach RTG: `Id_badania`, `Nazwa_badania`, `Cena_badania`.

Przykład

```
Id_badanie;Nazwa_badania;Cena_Badania
Id_1;RTG kosci pietowej;30
Id_10;RTG zeber;60
Id_11;RTG obojczyka;50
```

Wykorzystując dane zawarte w tych plikach i dostępne narzędzia informatyczne, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_RTG.txt`. Odpowiedź do każdego zadania poprzedź numerem je oznaczającym.

103.1.

Podaj nazwisko, imię oraz rok urodzenia pacjenta, któremu wykonano najwięcej zdjęć RTG, oraz liczbę tych zdjęć. Skorzystaj z informacji, że dwie pierwsze cyfry numeru PESEL to rok urodzenia.

103.2.

Pacjentami SOR byli mieszkańcy 16 oddziałów NFZ. Oddziały te pokrywają koszty wykonanych zdjęć RTG osób, które do nich należą. Utwórz zestawienie zawierające informacje o sumie kosztów zdjęć RTG poniesionych przez poszczególne oddziały NFZ. Zestawienie posortuj malejąco ze względu na sumę kosztów.

103.3.

Podaj rodzaj zdjęć RTG, które wykonano:

a) najwięcej razy,

b) u największej liczby pacjentów.

W odpowiedzi do punktu a) podaj też, ile razy wykonano dane badanie. W odpowiedzi do punktu b) podaj też liczbę pacjentów, u których wykonano dane badanie.

103.4.

Podaj liczbę kobiet oraz liczbę mężczyzn, którym wykonano zdjęcia RTG.

103.5.

Dla okresu od roku 1900 do 1999 utwórz zestawienie zawierające numery kolejnych dziesięcioleci oraz liczbę pacjentów urodzonych w tych dziesięcioleciach. Przyjmujemy, że pierwsze dziesięciolecie to okres 1900-1909, drugie dziesięciolecie to okres 1910–1919 itd.

Zadanie 104.
Wiązka zadań Leki refundowane

Pracownik NFZ ma za zadanie skontrolować recepty na leki refundowane wypisane w jednej z przychodni w ciągu pierwszego kwartału 2015 roku. Ma do dyspozycji trzy pliki tekstowe: `recepty.txt`, `leki_refundowane.txt` i `grupy_lekow.txt`[3].

Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są średnikami.

Plik o nazwie `recepty.txt` zawiera informacje dotyczące wypisywanych recept; w każdym wierszu znajduje się: identyfikator recepty (`ID_recepty`), data wypisania recepty (`Data`) oraz 13-cyfrowy kod wypisanego leku (`Kod_leku`). Jedna recepta może zawierać maksymalnie pięć leków.

Przykład

```
ID_recepty;Data;Kod_leku
11/2015;2015-01-02;5909990752720
12/2015;2015-01-02;5909990969753
12/2015;2015-01-02;5909990967247
```

W pliku `leki_refundowane.txt` zapisane są w każdym wierszu: 13-cyfrowy kod leku (`Kod_leku`), maksymalnie 90-znakowa nazwa substancji czynnej (`Subst_czynna`), maksymalnie 160-znakowa nazwa leku, zawierająca postać i dawkę leku, (`Nazwa`), identyfikator

[3] Źródło: http://www.mz.gov.pl/leki/refundacja/lista-lekow-refundowanych-obwieszczenia-ministra-zdrowia

grupy, do której dany lek należy (`Id_grupy`), jego cena (`Cena_detaliczna`) oraz cena refundowana (`Cena_refundowana`).

Przykład

```
Kod_leku;Subst_czynna;Nazwa;Id_grupy;Cena_detaliczna;Cena_refundowana
4013054024331;Metforminum;Siofor 500, tabl. powl., 500 mg;15.0;12,21;3,2
4013054024348;Metforminum;Siofor 850, tabl. powl., 850 mg;15.0;19,87;4,08
5901720140012;Doxazosinum;Dozox, tabl., 4 mg;76.0;70,74;9,6
```

Plik o nazwie `grupy_lekow.txt` zawiera identyfikator grupy (`Id_grupy`) oraz maksymalnie 200-znakową nazwę grupy (`Nazwa_grupy`).

Przykład

```
Id_grupy;Nazwa_grupy
1.0;Leki blokujace receptory histaminowe H2 - stosowane doustnie
10.0;Leki przeciwbiegunkowe - loperamid
100.1;Sulfametoksazol w polaczeniu z trimetoprymem do stosowania doustnego
- postacie stale
```

Korzystając z danych zawartych w tych plikach oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź zamieść w pliku `wyniki.txt`, poprzedzając ją numerem odpowiedniego zadania.

104.1.

Podaj, w którym dniu kontrolowanego okresu wypisano w przychodni najwięcej recept, podaj odpowiednią datę i liczbę recept z tego dnia. Pamiętaj, że na jednej recepcie może być wypisanych kilka leków.

104.2.

Podaj nazwę grupy leków, w której znajduje się najwięcej leków refundowanych w 100%, czyli takich, których cena refundowana wynosi 0 zł.

104.3.

Sporządź zestawienie, w którym dla każdego miesiąca podana będzie liczba wszystkich wypisanych recept oraz sumaryczna wartość wszystkich leków z recept wypisanych w tym miesiącu (w obliczeniach weź pod uwagę cenę detaliczną).

104.4.

Leki mają różne ceny detaliczne. Podaj cenę detaliczną najdroższego leku, na który wypisano receptę w badanej przychodni, oraz nazwę grupy, do której ten lek należy.

104.5.

Dopłata funduszu do leku to różnica pomiędzy ceną detaliczną a ceną refundowaną.

Utwórz zestawienie zawierające daty wystawienia i identyfikatory recept, dla których suma dopłat do wszystkich leków z danej recepty jest większa niż 2000 zł. Zestawienie posortuj rosnąco według dat.

Zadanie 105.

Wiązka zadań *Rośliny ogrodowe*

W poniżej opisanych plikach zgromadzono dane dotyczące zamówień złożonych w internetowym sklepie ogrodniczym[4]. Zakupów mogą dokonywać klienci, którzy dokonali wcześniejszej rejestracji w sklepie. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są średnikami.

W pliku `osoby.txt` każdy wiersz zawiera informacje o zamawiającym: identyfikator klienta (`Id_klienta`), jego nazwisko (`Nazwisko`), imię (`Imie`) i miasto, z którego on pochodzi (`Miasto`).

Przykład

```
Id_klienta;Nazwisko;Imie;Miasto
200;Koprowski;Maurycy;Wroclaw
```

W pliku `rosliny.txt` zawarte są informacje o roślinach ogrodowych dostępnych w sklepie internetowym. Każdy wiersz zawiera następujące informacje: identyfikator rośliny (`Id_rosliny`), nazwę rośliny (`Nazwa`), cenę jej sadzonki (`Cena`), kolor kwiatów (`Kolor_kwiatow`), okres kwitnienia (`Okres_kwitnienia`), rozmiary doniczki (`Rozmiary_doniczki`).

Przykład

```
Id_rosliny;Nazwa;Cena;Kolor_kwiatow;Okres_kwitnienia;Rozmiary_doniczki
54;Aconitum napellus;5;ciemno-niebieskie;IX-X;13x13x13
```

W pliku `zamowienia.txt` każdy wiersz zawiera informacje o pojedynczej pozycji zamówienia: numer porządkowy (`Lp`), identyfikator klienta (`Id_klienta`), datę (`Data`), liczbę sadzonek (`Liczba_sadzonek`) i identyfikator rośliny (`Id_rosliny`).

Przykład

```
Lp;Id_klienta;Data;Liczba_sadzonek;Id_rosliny
1;546;2014-03-01;6;554
```

Wykorzystując dane zawarte w plikach `osoby.txt`, `rosliny.txt`, `zamowienia.txt`, wykonaj poniższe polecenia. Odpowiedzi zapisz do pliku `wyniki_rosliny.txt`, a każdą z nich poprzedź cyfrą oznaczającą odpowiednie zadanie.

105.1.

Na zakończenie dnia sklep przygotowywał dla każdego klienta specyfikację zamówionego w ciągu całego dnia towaru i na tej podstawie sporządzał fakturę, która była dołączana do wysyłki (kumulowanie zakupów od jednego klienta zmniejsza liczbę przesyłek kurierskich). To oznacza, że na jednej fakturze były umieszczane pozycje z zamówień złożonych tego samego dnia. Wartość faktury to suma wartości zakupionych towarów.

Przykład

Klient zakupił w danym dniu 6 sadzonek w cenie 9 zł oraz 15 sadzonek w cenie 10 zł. Wartość jego faktury wynosi $6 \cdot 9 + 15 \cdot 10 = 204$ zł.

---

[4] Dane o roślinach zostały przygotowane na podstawie http://ogrodkomercyjny.pl/

Oblicz, ile faktur wystawił sklep ogrodniczy, oraz podaj najwyższą wartość wystawionej faktury. Podaj też, ilu klientom wystawiono więcej niż jedną fakturę.

105.2.

Utwórz zestawienie, w którym podasz liczbę pozycji złożonych przez osoby z poszczególnych miast zamówień na rośliny kwitnące dokładnie w okresie VII–VIII.

105.3.

Wyszukaj pozycje zamówień, które dotyczą więcej niż 10 sadzonek roślin o kwiatach w kolorze `bialo-liliowe`. Dla znalezionych zamówień podaj nazwiska i imiona osób, które je złożyły, liczbę sadzonek oraz nazwy roślin.

105.4.

Podaj nazwy roślin, które nie zostały uwzględnione w żadnym zamówieniu.

105.5.

Utwórz zestawienie zawierające liczbę pozycji zamówień roślin w poszczególnych rozmiarach doniczek.

Zadanie 106.

Wiązka zadań *Obserwacje ptaków*

Wolontariusze zarejestrowani w serwisie internetowym *www.awibaza.pl* wprowadzają do bazy informacje o swoich obserwacjach ptaków. W trzech plikach tekstowych przedstawiono dane zaczerpnięte z tej bazy. Dane w wierszach oddzielone są pojedynczymi znakami tabulacji. W każdym pliku pierwszy wiersz jest wierszem nagłówkowym

Plik `gatunki.txt` zawiera informacje o gatunkach ptaków: identyfikatory `ID_gatunku`, nazwy zwyczajowe (`nazwa_zwyczajowa`) i łacińskie (`nazwa_lacinska`). Każdy gatunek opisany jest w osobnym wierszu.

Przykład

```
ID_gatunku  nazwa_zwyczajowa     nazwa_lacinska
5           bazant zlocisty      Chrysolophus pictus
17          bielik               Haliaeetus albicilla
54          gawron               Corvus frugilegus
```

W pliku `lokalizacje.txt` jest lista miejsc, w których dokonywano obserwacji. W każdym miejscu można było dokonywać wielu obserwacji w różnych terminach. W pliku podano dla poszczególnych miejsc: identyfikator miejsca (`ID_lokalizacji`), jego nazwę (`lokalizacja`), nazwę powiatu, na terenie którego się ono znajduje (`powiat`), oraz jego krótki opis (`opis`).

```
ID_lokalizacji  lokalizacja     powiat       opis
1               Cieplewo        gdanski      laka podmokla, jezioro
6               Hel             pucki        plaza, brzeg zatoki
9               Koscierzyna     koscierski   transekt przez miasto
```

W pliku `obserwacje.txt` w każdym wierszu znajduje się zapis danych z jednej obserwacji. Każda obserwacja dotyczyła jednego gatunku ptaków, ale mogła obejmować wiele osobników tego gatunku. Dla każdej obserwacji podano: identyfikator obserwowanego gatunku

(`ID_gatunku`), identyfikator lokalizacji obserwacji (`ID_lokalizacji`), datę i czas jej początku (`poczatek`), datę i czas jej zakończenia (`koniec`), liczebność ptaków (`liczebnosc`) oraz ich zachowanie w czasie obserwacji (`zachowanie`). Obserwacja zaczynała się i kończyła zawsze w tym samym dniu.

```
ID_gatunku  ID_lokalizacji  poczatek          koniec            liczebnosc  zachowanie
92          24              1984-12-16 12:55  1984-12-16 13:55  107           plywa
124         13              2008-01-14 08:15  2008-01-14 11:45    1       odpoczywa
23           3              2014-10-25 07:00  2014-10-25 07:15    4          zeruje
```

Rozwiąż poniższe zadania, wykorzystując dostępne narzędzia informatyczne. Wyniki zamieść w pliku tekstowym o nazwie `ptaki_wyniki.txt`. Do oceny oddaj plik tekstowy zawierający wynik zadań oraz plik zawierający komputerową realizację twojego rozwiązania.

106.1.

Podać nazwy zwyczajowe trzech gatunków ptaków, które były obiektem największej liczby obserwacji.

106.2.

W których miesiącach można było zaobserwować remiza? Podaj numery miesięcy (liczby z zakresu 1..12), w których obserwowano gatunek o nazwie zwyczajowej „remiz". Dla każdego z tych miesięcy podaj łączną liczbę wszystkich zaobserwowanych osobników remiza. Weź pod uwagę wszystkie obserwacje, z różnych lat.

106.3.

Które z europejskich gatunków krukowatych — ptaków z rodzaju *Corvus* — można zaobserwować w miastach? Ile ich zaobserwowano?

Podaj nazwy zwyczajowe wszystkich gatunków ptaków z rodzaju *Corvus* (słowo *Corvus* stanowi fragment łacińskiej nazwy gatunku) zaobserwowanych w lokalizacjach położonych na terenach miejskich (zawierających słowo "miasto" w opisie lokalizacji obserwacji).

106.4.

Obserwacje dokonane w tej samej lokalizacji i tym samym czasie (mające jednakowe wartości w polu `poczatek` i w polu `koniec`) stanowią **grupę**.

a) Wyznacz **grupę obserwacji trwającą najdłużej.** Podaj jej lokalizację, datę, czas trwania w minutach i łączną liczbę osobników zaobserwowanych w tej grupie.

Sprawnością grupy obserwacji nazywamy średnią liczbę osobników wszystkich gatunków obserwowanych w ciągu 1 minuty przez grupę.

$$\text{sprawność} = \frac{\text{łączna liczba osobników obserwowanych w grupie}}{\text{czas trwania obserwacji grupy w minutach}}$$

b) Podaj **największą osiągniętą sprawność** grupy obserwacji, w zaokrągleniu do 3 cyfr po przecinku, oraz datę i lokalizację grupy, która ją osiągnęła.

106.5.

Wyszukaj wszystkie obserwacje osobników **żurawia** (o nazwie zwyczajowej: „zuraw").

a) Podaj liczbę wszystkich zaobserwowanych osobników żurawia.

b) Utwórz w postaci tabeli 2-wymiarowej zestawienie, w którym dla poszczególnych powiatów podasz liczby osobników żurawia zaobserwowanych na ich terenach, z podziałem na różne zachowania obserwowanego ptaka (*gniazduje*, *leci* itp.).

Zadanie 107.

Wiązka zadań *Loty pasażerskie*

W plikach: `loty.txt, pasazerowie.txt, bilety.txt` znajdują się informacje na temat lotów, pasażerów i biletów lotniczych zakupionych przez nich w biurze podróży w drugim kwartale 2014 roku. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

W pliku `loty.txt` znajduje się 1027 wierszy z informacjami o lotach pasażerskich: numerem identyfikacyjnym (`id_lotu`), miejscem docelowym (`miejsce_docelowe`), datą wylotu (`data`) oraz godziną wylotu (`godzina`).

Przykład

```
id_lotu   miejsce_docelowe    data          godzina
37        Warszawa            2014-04-04    12:05
38        Zurych              2014-04-04    13:50
39        Londyn Stansed      2014-04-04    18:10
```

Plik o nazwie `pasazerowie.txt` zawiera 302 wiersze z informacjami na temat pasażerów, którzy kupili bilety. Są to: identyfikator pasażera (`id_pasazera`), jego nazwisko (`nazwisko`), imię (`imie`), ulica, przy której mieszka, wraz z numerem domu (`adres`), miejscowość (`miejscowosc`) oraz telefon (`tel`).

Przykład

```
id_pasazera     nazwisko   imie  adres       miejscowosc     tel
202   Antczak   Edyta Czerwcowa 40/6        Walbrzych       735223964
203   Karpik    Hanna Drewniana 8/6         Dzierzoniow     312271637
```

W pliku `bilety.txt` znajduje się 2251 wierszy z informacjami na temat zakupionych przez pasażerów biletów: numerem identyfikacyjnym lotu (`id_lotu`) oraz identyfikatorem pasażera (`id_pasazera`).

Przykład

```
id_lotu   id_pasazera
142  100
161  420
170  161
171  155
```

Wykorzystując dane zawarte w tych plikach i dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie `wyniki_loty_pasazerskie.txt`. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

107.1.

Podaj 3 miejsca docelowe, do których wyloty odbyły się w największą liczbę dni. W zestawieniu podaj miejsce docelowe oraz liczbę dni.

107.2.

Utwórz zestawienie, w którym podasz nazwiska i imiona pasażerów, którzy zakupili więcej niż 15 biletów, oraz liczbę biletów kupionych przez każdego z tych pasażerów.

107.3.

Utwórz zestawienie, w którym dla każdego numeru miesiąca z badanego okresu podasz liczbę biletów kupionych przez osoby z Wrocławia (`Wroclaw`).

107.4.

Utwórz zestawienie, w którym podasz imiona i nazwiska pasażerów, którzy kupili bilety do dowolnego lotniska w Londynie na samoloty, których wyloty odbyły się między 8:00 a 10:00. Zestawienie posortuj rosnąco według kolejności alfabetycznej nazwisk.

107.5.

Utwórz zestawienie lotów, na które nie kupiono w biurze biletów. Podaj w nim numery lotów i ich miejsca docelowe.

Zadanie 108.

Wiązka zadań *Stacje benzynowe*

Informacje o wybranych drogach w Polsce i położonych przy nich stacjach benzynowych zamieszczone są w plikach: `drogi.txt`, `kategorie.txt`, `sieci.txt` oraz `stacje.txt`. Dane w plikach rozdzielone są znakiem tabulatora, pierwszy wiersz każdego pliku jest wierszem nagłówkowym.

Plik `drogi.txt` zawiera informacje dotyczące dróg. Są to: unikalny numer drogi (id_drogi), nazwa drogi (nazwa), jej długość (dlugosc) oraz identyfikator jej kategorii (id kategorii). Długość drogi jest podana w kilometrach i zaokrąglona w dół do pełnych kilometrów.

Przykład

```
id_drogi    nazwa dlugosc     id_kategorii
1     Autostrada Bursztynowa  582   A
2     Autostrada Wolnosci     623   A
3     Zachodnia Droga Ekspresowa    480   S
```

Plik `kategorie.txt` zawiera informacje dotyczące kategorii dróg: unikalny identyfikator kategorii (id_kategori) oraz nazwę kategorii (kategoria).

Przykład

```
id_kategorii      kategoria
A     autostrada
S     droga ekspresowa
```

Plik `sieci.txt` zawiera informacje dotyczące sieci stacji benzynowych: unikalny identyfikator sieci (id_sieci) oraz nazwę sieci, do której należy stacja (nazwa_sieci).

Przykład

```
id_sieci    nazwa_sieci
1     Dobre Paliwo
2     Standard Oil
```

Plik `stacje.txt` zawiera informacje dotyczące stacji benzynowych: unikalny identyfikator stacji (id_stacji), numer drogi, przy której jest położona stacja (id_drogi), oraz identyfikator sieci, do której ona należy stacja (id_sieci).

Przykład

```
id_stacji   id_drogi    id_sieci
1     8     8
2     8     5
```

Korzystając z danych zawartych w tych plikach oraz dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Odpowiedzi zapisz w pliku `wyniki.txt`, a odpowiedź do każdego zadania poprzedź numerem oznaczającym to zadanie.

108.1.

Podaj sumaryczną długość dróg wszystkich kategorii.

108.2.

Podaj nazwę i długość najdłuższej drogi, przy której nie leży żadna stacja benzynowa.

108.3.

Średnia odległość pomiędzy stacjami benzynowymi to stosunek całkowitej długości drogi do liczby położonych przy niej stacji. Jeżeli przy drodze nie ma stacji benzynowych, odległość ta jest niezdefiniowana i danej drogi nie bierzemy pod uwagę podczas porównywania średnich odległości pomiędzy stacjami.

Znajdź drogę, dla której średnia odległość między stacjami benzynowymi jest najmniejsza. Podaj numer drogi, jej nazwę oraz średnią odległość pomiędzy stacjami na tej drodze, z dokładnością do 0,1 km.

108.4.

Podaj nazwy dróg mających w nazwie słowo „autostrada”, ale będących drogami innych kategorii.

108.5.

Wykonaj zestawienie liczby stacji benzynowych należących do poszczególnych sieci z podziałem na kategorie dróg, przy których są położone te stacje.

Zadanie 109.

Wiązka zadań *Urządzenia budowlane*

Constall jest liderem wśród firm wynajmujących urządzenia i rusztowania dla budownictwa. W trzech plikach tekstowych `sprzet_budowlany.txt`, `klienci.txt` i `wynajem.txt` zostały zapisane dane dotyczące działalności firmy w 2014 roku.

Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

Plik o nazwie `sprzet_budowlany.txt` zawiera informacje dotyczące oferowanego sprzętu budowlanego. W każdym wierszu znajduje się: identyfikator sprzętu (`ID_sprzetu`), nazwa sprzętu (`Nazwa_sprzetu`), koszt jego wynajęcia na dobę podany w zł (`Koszt_wynajecia`), kaucja za sprzęt w zł (`Kaucja`).

Przykład

```
ID_sprzetu  Nazwa_sprzetu                      Koszt_wynajecia   Kaucja
1           Agregat hydrauliczny ATLAS         100               1200
2           Agregat jednofazowy-moc: 2,8-3,2kW 65                1000
```

W pliku `klienci.txt` zapisane są w każdym wierszu: numer dowodu osobistego osoby wynajmującej sprzęt (`Nr_dowodu_osoby`), imię tej osoby (`Imie`) oraz jej nazwisko (`Nazwisko`).

Przykład

```
Nr_dowodu_osoby     Imie      Nazwisko
XGF208075           Radoslaw  Warszawski
GUZ058053           Kacper    Szwaja
```

Plik o nazwie `wynajem.txt` zawiera: datę wypożyczenia sprzętu (`Data_wypozycz`), datę jego zwrotu (`Data_zwrotu`), identyfikator wypożyczanego sprzętu (`ID_sprzetu`), odległość od magazynu firmy do miejsca, w którym sprzęt ma być odebrany przez klienta, podaną w km (`Transport_km`), numer dowodu osobistego klienta (`Nr_dowodu_osoby`).

Przykład

```
Data_wypozycz     Data_zwrotu ID_sprzetu  Transport_km      Nr_dowodu_osoby
2014-01-02        2014-01-07  106         0                 MZM066623
2014-01-02        2014-01-07  14          0                 FXN638961
```

Firma Constall pobiera opłaty za usługę transportu urządzenia do klienta według poniższego taryfikatora:

- transport do 10 km włącznie — 50 zł,
- transport powyżej 10 km — 100 zł.

Korzystając z danych zawartych w tych plikach oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź umieść w pliku `wyniki.txt`, poprzedzając ją numerem odpowiedniego zadania.

109.1.

Podaj nazwę urządzenia wypożyczanego najczęściej oraz liczbę wypożyczeń tego urządzenia.

109.2.

Jaką łączną kwotę kaucji za wszystkie wypożyczone przez siebie urządzenia wpłacił Andrzej Rydawski identyfikujący się dowodem osobistym o numerze JCK343973?

109.3.

Podaj imiona i nazwiska osób, które wynajmowały od firmy Constall w 2014 roku więcej niż trzy różne urządzenia.

109.4.

Podaj liczbę zarejestrowanych w bazie firmy klientów, którzy w 2014 roku nie wypożyczyli żadnego urządzenia.

109.5.

Dla każdego miesiąca (od stycznia do grudnia) podaj:

a) przychód firmy z tytułu wypożyczeń urządzeń budowlanych w tym miesiącu,

b) przychód z tytułu opłat za transport w tym miesiącu.

Przykład do punktu a)

Jeżeli drabina aluminiowa 3-częściowa została wypożyczona 10.12.2014, a zwrócona 13.12.2014, to koszt wynajęcia jest obliczany następująco:

`liczba dni * cena za dobę` (3*300zł=900 zł).

Nie ma możliwości wypożyczenia i oddania sprzętu w tym samym dniu.

Uwaga: Opłata jest pobierana w dniu wynajęcia urządzenia i należy ją uwzględnić jako przychód w miesiącu, w którym urządzenie zostało wypożyczone.

Zadanie 110.

Wiązka zadań *Miejscowości w Polsce*

Informacje o podziale administracyjnym Polski oraz o wszystkich miejscowościach w jej obrębie zapisane są w plikach: `wojewodztwa.txt`, `powiaty.txt`, `gminy.txt` oraz `miejscowosci.txt`. Dane w plikach rozdzielone są znakiem tabulatora, pierwszy wiersz każdego pliku jest wierszem nagłówkowym.

Plik `wojewodztwa.txt` zawiera informacje dotyczące województw: unikalny numer województwa (id_wojewodztwa) oraz nazwę województwa (nazwa_wojewodztwa).

Przykład

id_wojewodztwa	nazwa_wojewodztwa
02	dolnoslaskie
04	kujawsko-pomorskie

Plik `powiaty.txt` zawiera informacje dotyczące powiatów: unikalny numer powiatu (id_powiatu), numer województwa, do którego on należy (id_wojewodztwa), oraz nazwę powiatu (nazwa_powiatu).

Przykład

id_powiatu	id_wojewodztwa	nazwa_powiatu
0201	02	boleslawiecki
0202	02	dzierzoniowski

Plik `gminy.txt` zawiera informacje dotyczące gmin: unikalny numer gminy (id_gminy), numer powiatu, do którego ona należy (id_powiatu), oraz nazwę gminy (nazwa_gminy).

Przykład

id_gminy	id_powiatu	nazwa_gminy
0201022	0201	Boleslawiec
0206011	0206	Karpacz

Plik `miejscowosci.txt` zawiera informacje dotyczące miejscowości: unikalny numer miejscowości (id_miejscowosci), numer gminy, w której jest ona położona(id_gminy), nazwę miejscowości (nazwa_miejscowosci) oraz słowne określenie typu miejscowości (typ_miejscowosci).

Przykład

id_miejscowosci	id_gminy	nazwa_miejscowosci	typ_miejscowosci
0162398	2206032	Abisynia	osada
0954722	0609052	Abramowice Koscielne	wies

Wykorzystując dane zawarte w tych plikach oraz dostępne narzędzia informatyczne, rozwiąż poniższe zadania. Odpowiedzi zapisz w pliku `wyniki.txt`, a odpowiedź do każdego podpunktu poprzedź numerem oznaczającym dane zadanie.

Jeżeli w treści pytań użyte są pojęcia:

- **gmina miejska** — należy przez to rozumieć gminę, na terenie której jest dokładnie jedna miejscowość i jest to miasto,
- **gmina miejsko-wiejska** — należy przez to rozumieć gminę, na terenie której znajduje się kilka miejscowości, z których przynajmniej jedna jest miastem;
- **gmina wiejska —** należy przez to rozumieć gminę, na terenie której nie ma żadnego miasta;
- **miasto na prawach powiatu** — należy przez to rozumieć miasto, którego nazwa jest taka sama jak nazwa powiatu, na terenie którego się znajduje.

110.1.

Podaj liczbę miast na terenie Polski.

110.2.

Oblicz, ile jest miejscowości każdego typu w powiecie brodnickim (wieś, miasto, osada itd.).

110.3.

Znajdź powtarzające się w kraju nazwy powiatów. Podaj nazwy tych powiatów oraz nazwy województw, w których te powiaty się znajdują. Zestawienie posortuj alfabetycznie według nazwy powiatów. Powiaty o tych samych nazwach uporządkuj alfabetycznie według nazw województw.

110.4.

Ile jest gmin wiejskich w województwie kujawsko-pomorskim?

110.5.

Wykonaj zestawienie gmin miejskich o nazwach zaczynających się na „J”, podając nazwy powiatów i województw, w których są one położone.

Zadanie 111.

Wiązka zadań *Malware*

Malware Domain List to niekomercyjny projekt społecznościowy, w którym tworzona jest lista adresów stron internetowych i dokumentów stwarzających zagrożenie: trojanów, exploitów itp. Listę na bieżąco uzupełniają profesjonalni entuzjaści bezpieczeństwa w sieci.

W trzech plikach tekstowych przedstawiono dane zaczerpnięte z tej listy. Dane w wierszach oddzielone są pojedynczymi znakami tabulacji. W każdym pliku pierwszy wiersz jest wierszem nagłówkowym.

W pliku `malware.txt` znajdują się pozycje z tej listy, wybrane z okresu od stycznia 2014 do stycznia 2015 roku włącznie. Podano: datę rejestracji zagrożenia (`data`), adres IP komputera udostępniającego zagrożenie (`IP`), opis zagrożenia (`opis`), numer ASN sieci, do której ten komputer należy (`ASN`), ścieżkę dostępu do szkodliwej strony lub do zasobu (`URL`).

Przykład

data	IP	opis	ASN	URL
2014-12-17	62.76.74.228	Trojan.Downloader	51408	my-screenshot.net/
2014-12-04	31.41.218.232	CryptoLocker	42655	mysda24.com/f/pacchetto_38.zip
2014-11-25	89.218.31.11	Script.exploit	9198	zakonodatelstvo.kz/russ.html

W pliku `asn.txt` znajdują się informacje o sieciach komputerowych, zawierające m.in. informacje o numerze ASN (*Autonomic System Number*) — identyfikatorze sieci, wykorzystywanym w konfiguracji routerów. W pliku podano dla każdej sieci: numer ASN (`ASN`), internetowy identyfikator kraju (`ID_kraju`), nazwę organizacji regionalnej przydzielającej adres ASN (`region`) oraz nazwę firmy zarządzającej siecią (`siec`).

Przykład

ASN	ID_kraju	region	siec
1267	it	ripencc	ASN-INFOSTRADA WIND Telecomunicazioni S.p.A.
2514	jp	apnic	INFOSPHERE NTT PC Communications, Inc
2914	us	arin	NTT-COMMUNICATIONS-2914 - NTT America, Inc

Regionalne organizacje przydzielające adresy ASN obejmują:

- apnic (*Asia Pacific Network Information Centre*) — rejon Azji i Pacyfiku,
- arin (*American Registry for Internet Numbers*) — rejon Ameryki Północnej,
- lacnic (*Latin-American and Caribbean*) — rejon Ameryki Łacińskiej i Wysp Karaibskich,
- ripencc (*Réseaux IP Européens*) — rejon Europy, Bliskiego Wschodu i centralnej Azji,
- afrinic — rejon Afryki.

W pliku `kraje.txt` podano nazwy krajów (`kraj`) oraz ich 2-literowe identyfikatory internetowe (`ID_kraju`).

Przykład

kraj	ID_kraju
Australia	au
France	fr
Hungary	hu

Rozwiąż poniższe zadania, wykorzystując dostępne narzędzia informatyczne. Wyniki zamieść w pliku tekstowym o nazwie `malware_wynik.txt`. Do oceny oddaj plik tekstowy zawierający wyniki oraz plik zawierający realizację komputerową Twojego rozwiązania.

111.1.

Znajdź te pozycje złośliwego oprogramowania, których celem jest **phishing**, czyli wyłudzanie informacji od użytkownika (w polu: *opis* zawierają łańcuch znaków: *phish* lub *Phish*). Podaj listę zawierającą dla każdej pozycji:

- nazwę kraju, z którego pochodzi komputer udostępniający zagrożenie,
- opis zagrożenia (`opis`),
- pełną ścieżkę dostępu do szkodliwego pliku (`URL`).

111.2.

Znajdź pięć sieci, z których komputery udostępniły najwięcej pozycji złośliwego oprogramowania. Podaj zestawienie zawierające dla każdej takiej sieci: nazwę sieci, nazwę kraju, w którym znajduje się ta sieć, liczbę stron lub dokumentów zawierających złośliwe oprogramowanie oraz liczbę różnych adresów IP, z których to oprogramowanie udostępniono.

111.3.

Dla każdego wpisu na listę *malware* określ domenę, z której udostępniono szkodliwe oprogramowanie. Nazwę domeny stanowią znaki pola URL liczone kolejno od lewej aż do pierwszego wystąpienia znaku „/”, bez tego znaku. W każdym polu `URL` danych znak „/” występuje przynajmniej jeden raz.

a) Podaj liczbę domen, z których pochodzi szkodliwe oprogramowanie.

b) Serwer DNS można skonfigurować tak, aby odpowiadał kilkoma adresami IP dla jednej domeny. Wyszukaj wśród domen te pozycje, którym odpowiada więcej niż jeden adres IP. Podaj nazwy tych domen i odpowiadające im liczby różnych adresów IP.

111.4.

Sporządź w postaci tabeli zestawienie, w którym podasz liczbę zarejestrowanych pozycji złośliwego oprogramowania **w każdym miesiącu roku 2014**, w podziale na poszczególne regiony: apnic, arin, lacnic, ripencc, afrinic. W zestawieniu nie uwzględniaj danych ze stycznia 2015 r.

111.5.

Złośliwe oprogramowanie może być ukryte w plikach graficznych i uaktywniać się podczas wyświetlania obrazu, wykorzystując luki w programach odtwarzających obraz.

Znajdź wszystkie wpisane na listę *malware* pozycje informujące o złośliwym kodzie ukrytym w obrazach zapisanych w formatach: jpg i png (pole *URL* kończy się: „.jpg" lub „.png").

Sporządź w postaci tabeli zestawienie, w którym dla każdego, w którym znajdował się komputer udostępniający szkodliwy kod w plikach graficznych, podasz liczby tych plików, z podziałem na format jpg i png.

Zadanie 112.

Wiązka zadań *Kod EAN*

Kod EAN (*European Article Number*) jest kodem kreskowym powszechnie stosowanym na opakowaniach towarów będących przedmiotem handlu. Czytnik kodu przetwarza go na ciąg cyfr z zakresu 0..9.

Dla uproszczenia zadania (choć rzeczywistość wykracza poza te reguły) przyjęto, że:

- kody wszystkich towarów są 13-cyfrowe,
- pierwsze 3 cyfry kodu są oznaczeniem kraju, w którym zarejestrowany jest producent,
- następne 4 cyfry identyfikują oddział firmy producenta,
- kolejne 5 cyfr jest identyfikatorem towaru,
- ostatnia cyfra jest cyfrą kontrolną.

W pliku `towary.txt` zebrano dane o wybranych słodyczach, napojach, deserach i używkach dostępnych na polskim rynku (nieco zmodyfikowane na potrzeby zadania). Kody EAN towarów są unikatowe. Towary o tej samej nazwie, lecz w opakowaniach o różnych masach, posiadają różne kody EAN. Dla każdego towaru podano: nazwę (`nazwa`), jego rodzaj (`rodzaj`), masę towaru w gramach (`masa`), podatek VAT (`VAT`) oraz kod EAN (`EAN`). Każdy towar opisany jest w osobnym wierszu. Przykład:

```
nazwa          rodzaj       masa      VAT      EAN
TIKI TAKI      czekolada    100       0,23     5909102008735
KOPIEC KRETA   ciasto       410       0,05     5909437089003
NIC NAC'S      przekaska     40       0,08     4018077798818
```

W pliku `kraje.txt` znajdują się nazwy krajów (`kraj`), w których zarejestrowani są producenci towarów, oraz przydzielone tym krajom 3-cyfrowe kody (`kod_kraju`), będące pierwszymi trzema cyframi kodów EAN. Większość krajów posiada jeden kod, ale są kraje, którym przydzielono więcej kodów. Przykład:

```
kod_kraju  kraj
304        Francja
338        Francja
380        Bulgaria
```

W pliku `producenci.txt` znajdują się informacje o oddziałach firm producentów towarów. Dla każdego oddziału podano: jego 4-cyfrowy kod stosowany w kodzie EAN (`kod_oddzialu`), 3-cyfrowy kod kraju, w którym oddział został zarejestrowany (`kod_kraju`), nazwę firmy (`nazwa_firmy`) oraz lokalizację jej oddziału — nazwę miasta, ewentualnie z nazwą dzielnicy (`lokalizacja`). Niektóre firmy mają wiele oddziałów, które mogą być zarejestrowane w różnych krajach. Przykład:

```
kod_kraju   kod_oddziału   lokalizacja          nazwa_firmy
590         9020           Torun                Pacific
590         1480           Jankowice            Kraft-Foods
304         5140           Velizy-Villacoublay  Kraft-Foods
```

W każdym pliku pierwszy wiersz jest wierszem nagłówkowym. Dane w wierszach oddzielone są znakiem tabulacji.

Rozwiąż poniższe zadania, wykorzystując dostępne narzędzia informatyczne. Wyniki zamieść w pliku tekstowym o nazwie `ean_wyniki.txt`. Do oceny oddaj ten plik oraz komputerową realizację rozwiązania.

112.1.

Dla każdego **rodzaju** towarów podaj **liczbę różnych krajów**, z których te towary pochodzą.

112.2.

Dla wszystkich producentów obecnych na naszym rynku, a zarejestrowanych w Czechach, podaj nazwy ich firm oraz lokalizacje ich oddziałów.

112.3.

Utwórz zestawienie, w którym dla każdego rodzaju towaru podasz najmniejszą masę towaru tego rodzaju oraz nazwę towaru tego rodzaju o najmniejszej masie.

112.4.

Wyszukaj towary z rodzaju „baton” w opakowaniach o masie przekraczającej 300g. Dla każdego z nich podaj: nazwę towaru, jego masę oraz nazwę firmy i nazwę kraju, w którym zarejestrowano oddział producenta tego towaru.

112.5.

Każdy towar ma określoną stawkę podatku VAT. Sporządź zestawienie, w którym dla każdego kraju podasz liczbę towarów pochodzących z tego kraju, z podziałem na poszczególne stawki podatku VAT.

5. Wykaz umiejętności ogólnych i szczegółowych sprawdzanych zadaniami

Zadanie 1.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach;

Zadanie 2.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych

Zadanie 3.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania 11) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych

Zadanie 4.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.

Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5. 4 dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5.7 opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 5.11 opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych, np.: reprezentacja liczb w dowolnym systemie pozycyjnym, w tym, w dwójkowym i szesnastkowym,

Zadanie 5.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.6. ocenia własności rozwiązania algorytmicznego, np., efektywność działania 5.11. opisuje podstawowe algorytmy i stosuje b) algorytmy wyszukiwania i porządkowania (sortowania), np. (…) algorytmy sortowania ciągu liczb (…) przez wstawianie 5.16 opisuje własności algorytmów na podstawie ich analizy 5.18 oblicza liczbę operacji wykonywanych przez algorytm

Zadanie 6.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy wyszukiwania i porządkowania (sortowania); 18) oblicza liczbę operacji wykonywanych przez algorytm

Zadanie 7.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 6) ocenia własności rozwiązania algorytmicznego (kompute-

	rowego), np. zgodność ze specyfikacją, efektywność działania; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: b) algorytmy wyszukiwania i porządkowania (sortowania) 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 8.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach;

Zadanie 9.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów; 9) stosuje rekurencje w prostych sytuacjach problemowych; 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 10.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów;

	9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: c) algorytmy numeryczne, np.: — obliczanie wartości wielomianu za pomocą schematu Hornera 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 11.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na tekstach, w tym obliczanie wartości wyrażenia podanego w postaci odwrotnej notacji polskiej;

Zadanie 12.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania 11) opisuje podstawowe algorytmy i stosuje: e) algorytmy kompresji i szyfrowania

Zadanie 13.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 11) opisuje podstawowe algorytmy i stosuje: […] f) algorytmy geometryczne

Zadanie 14.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy badające własności geometryczne 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 15.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów; 11) opisuje podstawowe algorytmy i stosuje: […] a) reprezentacje liczb w dowolnym systemie; f) algorytmy badające właściwości geometryczne […]; 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 16.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 11) opisuje podstawowe algorytmy i stosuje: […] f) algorytmy geometryczne

Zadanie 17.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje […] sytuacje z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązania problemów; 16) opisuje własności algorytmów na podstawie ich analizy;

	17) ocenia zgodność algorytmu ze specyfikacją; 26) ocenia poprawność rozwiązania problemu na podstawie jego testowania;

Zadanie 18.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5. 4 dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5.7 opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 5.11 opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych, np.: — iteracyjna i rekurencyjna realizacja algorytmu Euklidesa, 5.12 projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; 5.16 opisuje własności algorytmów na podstawie ich analizy;

Zadanie 19.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych, w tym reprezentację liczb w dowolnym systemie pozycyjnym.

Zadanie 20.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformuło-

	wania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych, np.: — reprezentacja liczb w dowolnym systemie pozycyjnym, w tym, w dwójkowym i szesnastkowym,

Zadanie 21.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5. 4 dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5.7 opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 5.11c) opisuje podstawowe algorytmy i stosuje algorytmy numeryczne, np.: zastosowania schematu Hornera: szybkie podnoszenie do potęgi, 5.16 opisuje własności algorytmów na podstawie ich analizy; 5.18 oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 22.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 6) ocenia własności rozwiązania algorytmicznego (komputerowego), np. zgodność ze specyfikacją, efektywność działania; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: c) algorytmy numeryczne, np.: — obliczanie wartości wielomianu za pomocą schematu Hornera 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 23.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.2. stosuje podejście algorytmiczne do rozwiązania problemu 5.4. dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji 5.5. posługuje się podstawowymi technikami algorytmicznymi 5.11. opisuje podstawowe algorytmy i stosuje c) algorytmy numeryczne, np.: — obliczanie wartości pierwiastka kwadratowego 5.14. dobiera odpowiednie struktury danych do realizacji algorytmu 5.16 opisuje własności algorytmów na podstawie ich analizy

Zadanie 24.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5. 4 dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5.7 opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 5.8 posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów; 5.9 stosuje rekurencję w prostych sytuacjach problemowych; 5.11 opisuje podstawowe algorytmy i stosuje b) algorytmy wyszukiwania i porządkowania (sortowania); 5.16 opisuje własności algorytmów na podstawie ich analizy; 5.18 oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 25.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.2. stosuje podejście algorytmiczne do rozwiązania problemu 5.4. dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji 5.5. posługuje się podstawowymi technikami algorytmicznymi 5.11. opisuje podstawowe algorytmy i stosuje d) algorytmy na tekstach, np.: — sprawdzanie, czy dany ciąg znaków tworzy palindrom

	5.14. dobiera odpowiednie struktury danych do realizacji algorytmu

Zadanie 26.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na tekstach

Zadanie 27.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na tekstach, np. [...] wyszukiwanie wzorca w tekście;

Zadanie 28.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Zdający: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na tekstach;

Zadanie 29.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algoryt-

	micznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania 11) opisuje podstawowe algorytmy i stosuje e) algorytmy kompresji i szyfrowania, np.: — kody znaków o zmiennej długości

Zadanie 30.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] e) algorytmy kompresji i szyfrowania, np. szyfr Cezara

Zadanie 31.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] e) algorytmy kompresji i szyfrowania, np. szyfr przestawieniowy

Zadanie 32.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń:

	2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] e) algorytmy szyfrowania i kompresji

Zadanie 33.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji; 5) posługuje się podstawowymi technikami algorytmicznymi; 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na liczbach całkowitych, w tym reprezentację liczb w dowolnym systemie pozycyjnym. 12) dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne;

Zadanie 34.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych
Wymagania szczegółowe	1.1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;

Zadanie 35.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych
Wymagania szczegółowe	1.1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;

Zadanie 36.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera danych liczbowych.
Wymagania szczegółowe	2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Zadanie 37.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera danych liczbowych.
Wymagania szczegółowe	2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Zadanie 38.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera danych liczbowych.
Wymagania szczegółowe	2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Zadanie 39.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych. II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	1.1. Uczeń przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków; 4.2 określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń; 4.3. przetwarza obrazy i filmy, np.: zmienia rozdzielczość, rozmiar, model barw, stosuje filtry;

Zadanie 40.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.

Zadanie 41.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów.

Zadanie 42.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: b) algorytmy wyszukiwania i porządkowania (sortowania), 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 43.

Wymagania ogólne	V. Ocena zagrożeń i ograniczeń, docenianie społecznych aspektów rozwoju i zastosowań informatyki
Wymagania szczegółowe	7.1 Uczeń … opisuje najważniejsze elementy procesu rozwoju informatyki i technologii informacyjno-komunikacyjnych.

Zadanie 44.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.
Wymagania szczegółowe	1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń: 1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków;

Zadanie 45.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych
Wymagania szczegółowe	PP 1.1) opisuje podstawowe elementy komputera, jego urządzenia zewnętrzne i towarzyszące i ich działanie w zależności od wartości ich podstawowych parametrów, wyjaśnia współdziałanie tych elementów;

Zadanie 46.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.
Wymagania szczegółowe	1. Posługiwanie się komputerem i jego oprogramowaniem, korzystanie z sieci komputerowej. Uczeń:

	2) wyjaśnia funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne;

Zadanie 47.

Wymagania ogólne	Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej, komunikowanie się za pomocą komputera i technologii informacyjno komunikacyjnych.
Wymagania szczegółowe	1.2) wyjaśnia funkcje systemu operacyjnego i korzysta z nich; opisuje różne systemy operacyjne;

Zadanie 48.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych.
Wymagania szczegółowe	3) przedstawia warstwowy model sieci komputerowych, określa ustawienia sieciowe danego komputera i jego lokalizacji w sieci, (...)

Zadanie 49.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych
Wymagania szczegółowe	1.4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania

Zadanie 50.

Wymagania ogólne	I. Bezpieczne posługiwanie się komputerem i jego oprogramowaniem, wykorzystanie sieci komputerowej; komunikowanie się za pomocą komputera i technologii informacyjno-komunikacyjnych
Wymagania szczegółowe	1.4) zapoznaje się z możliwościami nowych urządzeń związanych z technologiami informacyjno-komunikacyjnymi, poznaje nowe programy i systemy oprogramowania

Zadanie 51.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń:

	5) opisuje mechanizmy związane z bezpieczeństwem danych: (...)

Zadanie 52.

Wymagania ogólne	II Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł opracowywanie za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	1.1) przedstawia sposoby reprezentowania różnych form informacji w komputerze: liczb, znaków, obrazów, animacji, dźwięków; 4.2) określa własności grafiki rastrowej i wektorowej oraz charakteryzuje podstawowe formaty plików graficznych, tworzy i edytuje obrazy rastrowe i wektorowe z uwzględnieniem warstw i przekształceń;

Zadanie 53.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje […] sytuacje problemowe z różnych dziedzin; 11) […] stosuje: a) algorytmy na liczbach całkowitych, np. reprezentacje liczb w systemie dwójkowym.

Zadanie 54.

Wymagania ogólne	III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5.14 Dobiera odpowiednie struktury danych do realizacji algorytmu, w tym struktury dynamiczne;

Zadanie 55.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 16) opisuje własności algorytmów na podstawie ich analizy; 18) oblicza liczbę operacji wykonywanych przez algorytm;

Zadanie 56.

Wymagania ogólne	V. Ocena zagrożeń i ograniczeń, docenianie społecznych aspektów rozwoju i zastosowań informatyki

Wymagania szczegółowe	7.4 Uczeń … omawia zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci

Zadanie 57.

Wymagania ogólne	V. Ocena zagrożeń i ograniczeń, docenianie społecznych aspektów rozwoju i zastosowań informatyki
Wymagania szczegółowe	7.4 Uczeń … omawia zagadnienia przestępczości komputerowej, w tym piractwo komputerowe, nielegalne transakcje w sieci

Zadanie 58.

Wymagania ogólne	II. […] opracowanie za pomocą komputera danych liczbowych. III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np. efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy badające właściwości geometryczne […]; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

Zadanie 59.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.11 opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych, np.: — rozkładanie liczby na czynniki pierwsze,

	— i inne 5.12 projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; 5.15 stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; 5.21 przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 5.22 sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; 5.23 stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 5.25 dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;

Zadanie 60.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;

Zadanie 61.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;

Zadanie 62.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algoryt-

	micznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania 11) opisuje podstawowe algorytmy i stosuje: a) algorytmy na liczbach całkowitych — reprezentacja liczb w dowolnym systemie pozycyjnym 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu

Zadanie 63.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.11 opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych, np.: — reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym i szesnastkowym, — sprawdzanie, czy liczba jest liczbą pierwszą, — i inne 5.12 projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; 5.15 stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; 5.21 przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 5.22 sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; 5.23 stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 5.25 dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;

Zadanie 64.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe;

	2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np. efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania […]; a) algorytmy na liczbach; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania

Zadanie 65.

Wymagania ogólne	II. […] opracowanie za pomocą komputera danych liczbowych. III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np. efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych […]; b) algorytmy wyszukiwania […]; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

Zadanie 66.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wy-

	korzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.11 opisuje podstawowe algorytmy i stosuje a) algorytmy na liczbach całkowitych, np.: — sprawdzanie, czy liczba jest liczbą pierwszą, f) algorytmy badające własności geometryczne, np. — sprawdzanie warunku trójkąta 5.12 projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; 5.15 stosuje zasady programowania strukturalnego i modularnego do rozwiązywania problemu; 5.21 przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 5.22 sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów; 5.23 stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 5.25 dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;

Zadanie 67.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: […] a) algorytmy na liczbach całkowitych, np.: — reprezentacja liczb w dowolnym systemie pozycyjnym, w tym w dwójkowym… — sprawdzanie, czy liczba jest liczbą pierwszą … f) konstrukcje rekurencyjne: binarny fraktal Fibonacciego

Zadanie 68.

Wymagania ogólne	II. […] opracowanie za pomocą komputera tekstów, danych liczbowych. III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algo-

	rytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np. efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania […]; d) algorytmy na tekstach; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania

Zadanie 69.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np. efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] b) algorytmy wyszukiwania […]; d) algorytmy na tekstach; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania

Zadanie 70.

Wymagania ogólne	III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.11 opisuje podstawowe algorytmy i stosuje c) algorytmy numeryczne, np.: — obliczanie pola obszarów zamkniętych, 5.21 przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 5.25 dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania;

Zadanie 71.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 8) posługuje się metodą „dziel i zwyciężaj” w rozwiązywaniu problemów; 9) stosuje rekurencję w prostych sytuacjach problemowych. 11) opisuje podstawowe algorytmy i stosuje: c) algorytmy numeryczne, np.: — wyznaczanie miejsc zerowych funkcji metodą połowienia, 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;

Zadanie 72.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformuło-

	wania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] d) algorytmy na tekstach 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;

Zadanie 73.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] e) algorytmy na tekstach 12) projektuje rozwiązanie problemu (realizację algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;

Zadanie 74.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5.Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi 11) opisuje podstawowe algorytmy i stosuje: d) algorytmy na tekstach, np.: — porządkowanie alfabetyczne, 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 25) dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania

Zadanie 75.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, sto-

	sowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: […] e) algorytmy szyfrowania i kompresji 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu;

Zadanie 76.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: e) algorytmy kompresji i szyfrowania, np.: szyfr przestawieniowy 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;

Zadanie 77.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 11) opisuje podstawowe algorytmy i stosuje e) algorytmy kompresji i szyfrowania np.: — kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana, — szyfr Cezara, — szyfr przestawieniowy, — szyfr z kluczem jawnym (RSA), — wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym,

Zadanie 78.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 11) opisuje podstawowe algorytmy i stosuje e) algorytmy kompresji i szyfrowania np.: kody znaków o zmiennej długości, np. alfabet Morse'a, kod Huffmana, szyfr Cezara, szyfr przestawieniowy, szyfr z kluczem jawnym (RSA), wykorzystanie algorytmów szyfrowania, np. w podpisie elektronicznym,

Zadanie 79.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania; 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy badające własności geometryczne 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu;

Zadanie 80.

Wymagania ogólne	II. […] opracowanie za pomocą komputera danych liczbowych. III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 4) dobiera efektywny algorytm do rozwiązania problemu; 5) posługuje się podstawowymi technikami algorytmicznymi; 6) ocenia własność rozwiązania algorytmicznego […], np.

	efektywność działania. 11) opisuje podstawowe algorytmy i stosuje: […] f) algorytmy badające właściwości geometryczne […]; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

Zadanie 81.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania 11) opisuje podstawowe algorytmy i stosuje: f) algorytmy badające własności geometryczne 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu

Zadanie 82.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	4.4. wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 83.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.3 formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 5.25.dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania

Zadanie 84.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych. III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5.Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 5.1 analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 5.2 stosuje podejście algorytmiczne do rozwiązywania problemu; 5.3 formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 5.25.dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania

Zadanie 85.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 86.

Wymagania ogólne	II. […] opracowanie za pomocą komputera tekstów, danych liczbowych. III. Rozwiązywanie problemów i podejmowanie decyzji […] z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. […] opracowanie za pomocą komputera tekstów, danych liczbowych. Zdający: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5. Rozwiązywanie problemów i […] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych; 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu; 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania, instrukcje iteracyjne i warunkowe, rekurencję, funkcje i procedury, instrukcje wejścia i wyjścia, poprawnie tworzy strukturę programu; 26) ocenia poprawność komputerowego rozwiązania problemu na podstawie jego testowania.

Zadanie 87.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń:

	2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania

Zadanie 88.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: rysunków, tekstów, danych liczbowych, animacji, prezentacji multimedialnych i filmów. 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5. Rozwiązywanie problemów i podejmowanie decyzji, stosowanie podejścia algorytmicznego. Uczeń: 2) stosuje podejście algorytmiczne do rozwiązywania problemu 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania

Zadanie 89.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5.Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 5.25.dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania

Zadanie 90.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: […] tekstów, danych liczbowych. Uczeń: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 91.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych III. Rozwiązywanie problemów, podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.
Wymagania szczegółowe	4.4.wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów 5.Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu; 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 5.25.dobiera właściwy program użytkowy lub samodzielnie napisany program do rozwiązywanego zadania

Zadanie 92.

Wymagania ogólne	II. [...] opracowywanie za pomocą komputera danych liczbowych. III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: […] tekstów, danych liczbowych. Uczeń: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 93.

Wymagania ogólne	II. [...] opracowywanie za pomocą komputera danych liczbowych. III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: […] tekstów, danych liczbowych. Uczeń: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 94.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: […] tekstów, danych liczbowych. Uczeń: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 95.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 4) znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin;

Zadanie 96.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 4) znajduje odpowiednie informacje niezbędne do realizacji projektów z różnych dziedzin;

Zadanie 97.

Wymagania ogólne	II. [...] opracowywanie za pomocą komputera danych liczbowych. III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	4. Opracowywanie informacji za pomocą komputera, w tym: […] tekstów, danych liczbowych. Uczeń: 4) wykorzystuje arkusz kalkulacyjny do obrazowania zależności funkcyjnych i do zapisywania algorytmów. 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego. Uczeń: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 2) stosuje podejście algorytmiczne do rozwiązywania problemu 3) formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera; 7) opracowuje i przeprowadza wszystkie etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania;

Zadanie 98.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1.projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2.2.stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3.tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 99.

Wymagania ogólne	II. […] przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera […] danych liczbowych […]. III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, […].
Wymagania szczegółowe	2. […] selekcjonowanie, przetwarzanie […] informacji, korzystanie z różnych źródeł […] informacji. Zdający: 1) projektuje relacyjną bazę danych […]; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integral-

	ność danych na poziomie pól, tabel, relacji; 5. Rozwiązywanie problemów […]. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 25) dobiera właściwy program użytkowy […] do rozwiązania zadania.

Zadanie 100.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1. projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3. tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 101.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji

Zadanie 102.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1. projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3. tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 103.

Wymagania ogólne	II. […] przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera […] danych liczbowych […]. III Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, […].
Wymagania szczegółowe	2. […] selekcjonowanie, przetwarzanie […] informacji, korzystanie z różnych źródeł […] informacji. Zdający: 1) projektuje relacyjną bazę danych […]; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji; 5. Rozwiązywanie problemów […]. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin; 25) dobiera właściwy program użytkowy […] do rozwiązania zadania.

Zadanie 104.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1.projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2.2.stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3.tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 105.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowanie za pomocą komputera danych liczbowych
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji

Zadanie 106.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Zadanie 107.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1. projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2.2. stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3. tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 108.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 109.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji.
Wymagania szczegółowe	2.1.projektuje relacyjną bazę danych z zapewnieniem integral-

	ności danych; 2.2.stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych; 2.3.tworzy aplikację bazodanową, w tym sieciową, wykorzystując język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 110.

Wymagania ogólne	II. Wyszukiwanie, gromadzenie i przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, tekstów, danych liczbowych, motywów, animacji, prezentacji multimedialnych.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL); 3) tworzy aplikację bazodanową, w tym sieciową, wykorzystującą język zapytań, kwerendy, raporty; zapewnia integralność danych na poziomie pól, tabel, relacji;

Zadanie 111.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);

Zadanie 112.

Wymagania ogólne	III. Rozwiązywanie problemów i podejmowanie decyzji z zastosowaniem podejścia algorytmicznego.
Wymagania szczegółowe	2. Wyszukiwanie, gromadzenie, selekcjonowanie, przetwarzanie i wykorzystywanie informacji, współtworzenie zasobów w sieci, korzystanie z różnych źródeł i sposobów zdobywania informacji. Uczeń: 1) projektuje relacyjną bazę danych z zapewnieniem integralności danych; 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL);