

## Informatyka

Egzamin maturalny z informatyki odbył się w całym kraju w dniu 15 maja 2012 r. Maturzyści mogli wybrać informatykę jako przedmiot dodatkowy na poziomie podstawowym lub rozszerzonym.

### Opis arkuszy egzaminacyjnych

Arkusze egzaminacyjne z informatyki zostały opracowane na dwóch poziomach:

- podstawowym – *Arkusz I* (MIN-P1\_1P-122) oraz *Arkusz II* (MIN-P2\_1P-122)
- rozszerzonym – *Arkusz I* (MIN-R1\_1P-122) oraz *Arkusz II* (MIN-R2\_1P-122)

Egzamin na każdym poziomie składał się z dwóch części: pisemnej (*Arkusz I* – zadania rozwiązywane bez użycia komputera) oraz praktycznej (*Arkusz II* – zadania rozwiązywane z wykorzystaniem komputera).

Na poziomie podstawowym:

- *Arkusz I* zawierał 3 zadania, zdający mógł uzyskać za nie maksymalnie 20 punktów,
- *Arkusz II* zawierał 3 zadania praktyczne, za które zdający mógł uzyskać 30 punktów.

Egzamin trwał 75 minut w części I i 120 minut w części II.

Na poziomie rozszerzonym:

- *Arkusz I* zawierał 3 zadania, zdający mógł uzyskać za nie maksymalnie 20 punktów,
- *Arkusz II* zawierał 3 zadania praktyczne, za które zdający mógł uzyskać 30 punktów.

Egzamin trwał 90 minut w części I i 150 minut w części II.

Zadania w arkuszach sprawdzały wiadomości i umiejętności określone w standardach wymagań egzaminacyjnych, opisane w *Informatorze o egzaminie maturalnym od 2009 roku – Informatyka*.

### Poziom Podstawowy

#### *Arkusz I*

##### **Zadanie 1. Fibonacciego (7 pkt)**

Poniższa funkcja rekurencyjna *Fib* oblicza  $k$ -ty wyraz ciągu Fibonacciego.

*Dane:*  $k$  – liczba naturalna większa od zera

##### **Funkcja *Fib*( $k$ )**

1. Jeżeli  $k = 1$  lub  $k = 2$ , to wynikiem jest 1.
2. Jeżeli  $k > 2$ , to wynikiem jest  $Fib(k-1) + Fib(k-2)$ .

##### **Przykład:**

Zgodnie z powyższą definicją funkcji *Fib* mamy:

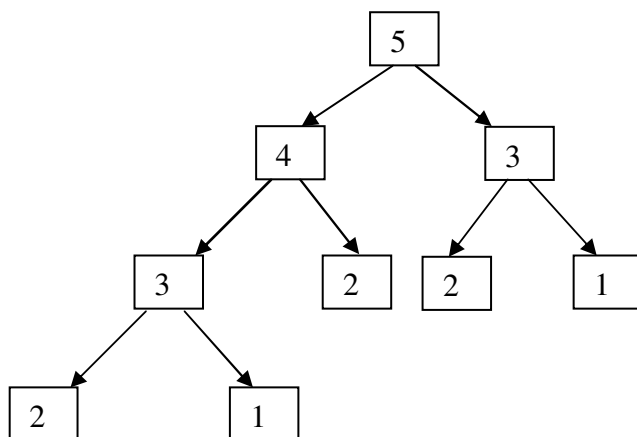
$$\begin{aligned} Fib(4) &= Fib(3) + Fib(2) = \\ &= [Fib(2) + Fib(1)] + Fib(2) = \\ &= [1 + 1] + 1 = 3 \end{aligned}$$

- a) Uzupełnij tabelę, wpisując dla podanych argumentów  $k$  wartości obliczane przez funkcję *Fib*.

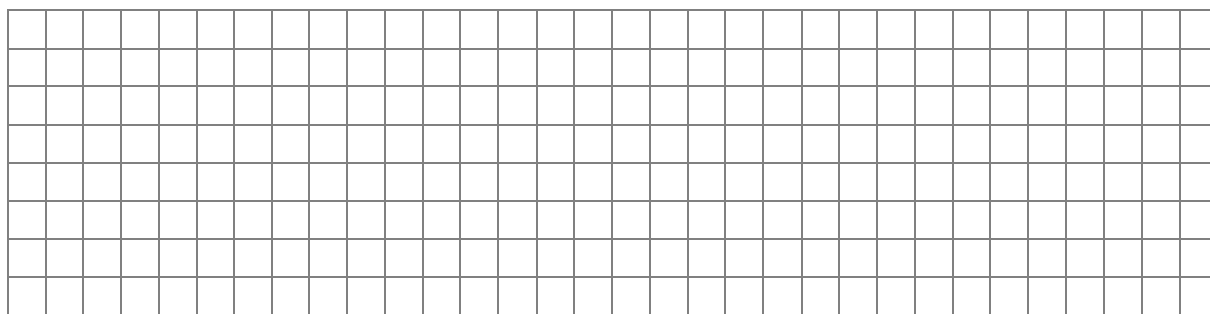
$k$	$Fib(k)$
1	1

2	1
3	2
...	...
8	
...	...
11	

- b) Wywołanie funkcji  $Fib(k)$  dla  $k > 2$  powoduje dwa kolejne wywołania tej funkcji z mniejszymi argumentami, które z kolei mogą wymagać kolejnych wywołań  $Fib$ , itd. Proces ten można zilustrować za pomocą tzw. drzewa wywołań rekurencyjnych. Poniżej prezentujemy drzewo wywołań rekurencyjnych dla  $k = 5$ . W węzłach drzewa znajdują się argumenty wywołań.



Narysuj drzewo wywołań rekurencyjnych dla  $Fib(6)$ .



- c)  $k$ -ty wyraz ciągu Fibonacciego można wyznaczyć iteracyjnie w następujący sposób:

*Dane:*  $k$  – liczba naturalna większa od zera

**Algorytm:**

1.  $Fi \leftarrow 1, Fi\_1 \leftarrow 1, i \leftarrow 2$

2. dopóki  $i < k$

$pom \leftarrow Fi$

$Fi \leftarrow Fi + Fi\_1$

$$i \leftarrow i + 1$$

Zdefiniujmy następujący ciąg:

- Pierwszy i drugi wyraz ciągu są równe 1.
- Jeśli  $k > 2$  i  $k$  jest parzyste, to  $k$ -ty wyraz jest sumą trzech wyrazów go poprzedzających.
- Jeśli  $k > 2$  i  $k$  jest nieparzyste, to  $k$ -ty wyraz jest równy wyrazowi o numerze  $(k-1)$ .

$k$	1	2	3	4	5	6	7	8
$k$ -ty wyraz	1	1	1	3	3	7	7	17

## Specyfikacja:

**Wynik:**  $k$ -ty wyraz ciągu zdefiniowanego powyżej

[illegible]

W zadaniu były badane umiejętności z I i II obszaru standardów. Zdający:

- zna podstawowe algorytmy i techniki algorytmiczne – rekurencję (I.7),
- wyodrębniania elementy składowe algorytmu (I.7),
- stosuje podstawowe algorytmy i struktury danych w rozwiązywaniu problemów informatycznych (II.5).

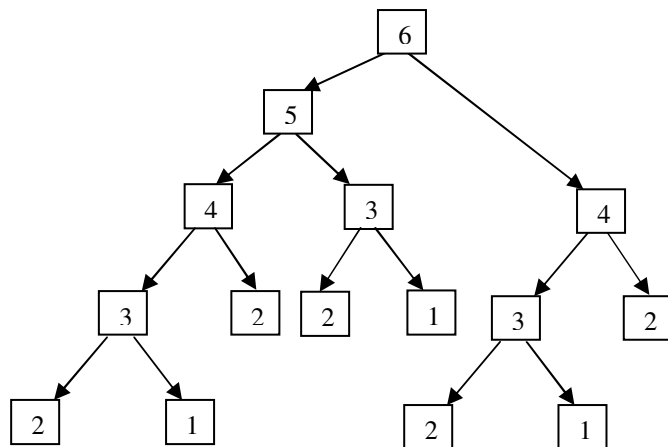
Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.a.	podaje dla wskazanych argumentów $k$ wartości obliczane przez funkcję	2	0,59	0,68	–	0,56
1.b.	rysuje drzewo wywołań rekurencyjnych.	1	0,90	0,95	–	0,89
1.c.	zapisuje w postaci iteracyjnej algorytm, który oblicza $k$ -ty wyraz ciągu.	4	0,23	0,21	–	0,24

**Przykładowy poprawny zapis rozwiązania:**

a)

$k$	$Fib(k)$
8	21
11	89

b)



1c.

rozwiązanie 1 $Fi \leftarrow 1, Fi\_1 \leftarrow 1, i \leftarrow 2$ dopóki  $i < k$ jeżeli  $(i+1) \bmod 2 = 0$  to $pom \leftarrow Fi$  $Fi \leftarrow Fi + Fi\_1 + Fi\_2$  $Fi\_2 \leftarrow Fi\_1$  $Fi\_1 \leftarrow pom$ 

w przeciwnym razie

 $Fi\_2 \leftarrow Fi\_1$  $Fi\_1 \leftarrow Fi$  $i \leftarrow i + 1$ wypisz  $Fi$ rozwiązanie 2 $Tab[1] \leftarrow 1, Tab[2] \leftarrow 1$  $i \leftarrow 3$ dopóki  $i < k$ jeżeli  $i \bmod 2 = 0$  to $Tab[i] \leftarrow Tab[i-3] + Tab[i-2] + Tab[i-1]$ 

w przeciwnym razie

 $Tab[i] \leftarrow Tab[i-1]$  $i \leftarrow i + 1$ wypisz  $Tab[i]$ **Komentarz:**

Zadanie pierwsze składa się z trzech podpunktów, w tym pierwsze dwa dotyczyły analizy funkcji rekurencyjnej  $Fib$ ; należało podać wartości zwracane przez funkcję dla podanych argumentów oraz narysować drzewo wywołań funkcji dla argumentu równego 6. Teoretycznie do rozwiązania punktu a) nie była konieczna analiza funkcji, jedynie znajomość definicji ciągu Fibonacciego. W punkcie c) zadaniem maturzysty było, dla zdefiniowanego ciągu, samodzielne zapisanie w postaci iteracyjnej algorytmu, który dla danej wartości  $k$  wyznacza  $k$ -ty wyraz tego ciągu. Była to najtrudniejsza część zadania, co znajduje odzwierciedlenie w punktacji (4 punkty, wobec 2 i 1 punktu za każdą z części a i b). Realizacja tego polecenia wymaga umiejętności rozróżnienia wyrazów parzystych i nieparzystych, obliczania wyróżnionych wyrazów parzystych i nieparzystych oraz zastosowania prawidłowej pętli, umożliwiającej obliczanie wszystkich elementów ciągu.

Wskaźnik łatwości 0,43 klasyfikuje to zadanie jako umiarkowanie trudne. O ile podanie prawidłowej odpowiedzi na pierwsze dwa podpunkty nie sprawiło zdającym większych problemów, to w przypadku zapisania algorytmu pojawiły się kłopoty. Niektórzy przepisywali algorytm ciągu Fibonacciego zamiast właściwego algorytmu, inni (nie

korzystający z tablicy) nie zapamiętywali wartości poprzedzających wyrazów ciągu, a część zdających przepisywała definicję algorytmu podaną w treści zadania.

### Zadanie 2. Diamenty (8 pkt)

W sejfie jubлера znajduje się  $n$  diamentów wycenionych odpowiednio na  $d_1, \dots, d_n$  złotych, przy czym żadne dwa diamenty nie są w tej samej cenie. Jubiler nie ujawnia cen diamentów, co oznacza, że tylko on zna ceny  $d_1, \dots, d_n$ .

Dla zainteresowanych klientów jubiler wykonuje operację porównania cen diamentów: dla wskazanych numerów  $i$  oraz  $j$  podaje, czy diament o numerze  $i$  ma wyższą cenę, niż diament o numerze  $j$ .

Przyjmijmy następujący sposób oznaczania wyniku operacji porównania cen:

$wi\acute{e}ksze(i, j) = \text{prawda}$ , gdy  $d_i > d_j$

$wi\acute{e}ksze(i, j) = \text{fa\lshsz}$ , gdy  $d_i < d_j$

a) Poniżej prezentujemy pewien algorytm korzystający z operacji porównania cen:

1.  $j \leftarrow 0$
2.  $i \leftarrow 1$
3. dopóki  $i < n$   
    jeżeli  $wi\acute{e}ksze(i, i+1)$  to  $j \leftarrow j+1$   
     $i \leftarrow i+1$
4. wypisz  $j$

Uzupełnij poniższą tabelę, podając wyniki działania powyższego algorytmu po jego wykonaniu dla wskazanych danych.

$n$	$d_1, \dots, d_n$	Wynik algorytmu
4	5 2 1 6	2
4	2 5 1 2	
4	1 2 3 4	
4	4 3 2 1	

b) Zapisz algorytm (w postaci listy kroków, schematu blokowego lub w wybranym języku programowania), który dla podanego ciągu cen diamentów znajduje numer diamentu o najwyższej cenie. W algorytmie zastosuj operację  $wi\acute{e}ksze$  porównania cen dwóch diamentów.

### Specyfikacja:

*Dane:*  $n$  – liczba naturalna większa od zera, oznaczająca liczbę diamentów

$d_1, \dots, d_n$  – ceny diamentów o kolejnych numerach  $1, 2, \dots, n$ ; ceny dwóch różnych diamentów są różne

*Wynik:*  $i$  – numer diamentu o najwyższej cenie

[illegible][illegible]

- zna podstawowe algorytmy i techniki algorytmiczne (I.7),
- stosuje podstawowe algorytmy i struktury danych w rozwiązywaniu problemów informatycznych (II.5).
- analizuje liczbę wykonywanych w algorytmie operacji (II.6),
- dobiera algorytm w celu rozwiązania problemu i zapisuje go w jednej z poniższej notacji: listy kroków, schematu blokowego, w języku programowania (II.5).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
2.a.	analizuje działanie algorytmu dla podanych danych.	3	0,69	0,78	–	0,65
2.b.	zapisuje algorytm wypisujący indeks największego elementu w podanym ciągu, analizuje liczbę wykonywanych w algorytmie operacji.	5	0,34	0,37	–	0,33

a)  
odp.: 1, 0, 3.

b)  
Przykładowe rozwiązanie:

- Drugie zadanie arkusza I sprawdzało umiejętność śledzenia i analizy algorytmu zapisanego w pseudokodzie, a także, podobnie do pierwszego, wymagało samodzielnego sformułowania

algorytmu. W treści zadania podany był algorytm, którego wynikiem działania jest liczba takich sytuacji, w których diament poprzedzający ma wyższą wartość niż diament następujący po nim. W punkcie a) należało prześledzić działanie algorytmu na czterech konkretnych wartościach diamentów. Podobnie do zadania pierwszego, najtrudniejszy okazał się podpunkt, w którym wymaga się samodzielnego sformułowania i zapisania algorytmu (łatwość wynosi 0,34). Algorytm powinien, dla podanego ciągu cen diamentów, znaleźć numer diamentu o najwyższej cenie. Należy zaznaczyć, że zadanie to nie powinno nastręczać dużych trudności, choćby dlatego że wyszukiwanie indeksu maksimum w ciągu liczb to jedno z klasycznych zadań w podstawowym kursie algorytmiki i programowania, a jego różne warianty już pojawiały się w zestawach maturalnych. Najczęstsze błędy pojawiały się w organizacji pętli sterującej oraz w myleniu ceny diamentu z jego pozycją (zwrot  $d_i$  zamiast  $i$ ). Rzadko pojawiały się rozwiązania polegające na sortowaniu bąbelkowym.

### Zadanie 3. Test (5 pkt)

W podpunktach a) – e) **zaznacz znakiem X** poprawne odpowiedzi.

**Uwaga:** W każdym podpunkcie poprawna jest tylko jedna odpowiedź.

Adres IP to 32-bitowa liczba zapisywana jako cztery binarne liczby ośmiobitowe oddzielone odstępami, bądź jako cztery liczby dziesiętne oddzielone kropkami. Na przykład:

```
10000000 00000001 00000010 11111110
128.1.2.254
```

to dwa różne zapisy tego samego adresu.

Poniżej podajemy dwie niepełne wersje tego samego adresu IP:

```
???????? 10101000 0000001 00000010
192.??? .1.2
```

gdzie znaki zapytania oznaczają brakujące cyfry.

- a) Która z poniższych liczb jest równa brakującej części powyższego adresu IP w postaci binarnej?
- 11000000
  - 10100000
  - 10111110
- b) Która z poniższych liczb jest równa brakującej części powyższego adresu IP w postaci dziesiętnej?
- 178
  - 168
  - 148
- c) Największa liczba dziesiętna, jaką można zapisać na 32 bitach jest
- równa 65 000.
  - większa od 1 123 000.
  - mniejsza od 4 000.
- d) Programowanie strukturalne to termin oznaczający
- tworzenie oprogramowania analizującego strukturę połączeń w sieci WWW.
  - programowanie nastawione na wykorzystanie struktury sprzętu, na którym uruchamiany będzie wynikowy program.
  - tworzenie programów zawierających struktury sterujące (np. pętle „dopóki”, „powtarzaj”, instrukcję „jeżeli”).

- e) Aby uniemożliwić odczytanie przez niepowołane osoby pliku przesyłanego pocztą elektroniczną, stosuje się narzędzia służące do
- archiwizacji.
  - kompilacji.
  - szyfrowania.

### **Sprawdzane umiejętności**

W zadaniu były badane umiejętności z I obszaru standardów. Zdający:

- zna sposoby reprezentowania informacji w komputerze (I.6),
- zna zasady programowania strukturalnego (I.8),
- zna narzędzia służące do zabezpieczania programów i danych w komputerze (I.3).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a.	dokonuje konwersji liczby binarnej na dziesiętną.	1	0,91	0,89	–	0,93
3.b.	dokonuje konwersji liczby dziesiętnej na binarną.	1	0,96	0,95	–	0,97
3.c.	określa długość binarnej reprezentacji w zależności od wielkości liczby.	1	0,77	0,74	–	0,79
3.d.	zna pojęcia związane z programowaniem strukturalnym.	1	0,94	0,86	–	0,97
3.e.	zna narzędzia służące do zabezpieczania programów i danych w komputerze.	1	1,00	1,00	–	1,00

### **Poprawny zapis rozwiązania:**

a1, b2, c2, d3, e3

### **Komentarz:**

Zadanie miało charakter testu wyboru, sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Zadanie okazało się łatwe, co świadczy o dobrym opanowaniu przez zadających treści sprawdzanych w zadaniu.



## Arkusz II

### Zadanie 4. Cyfry (10 pkt)

W kolejnych wierszach pliku `cyfry.txt` znajduje się 1000 liczb naturalnych, mniejszych niż  $10^9$  (jeden miliard), po jednej liczbie w każdym wierszu.

**Napisz program**, który da odpowiedzi do poniższych podpunktów. Każdą odpowiedź zapisz w pliku `zadanie4.txt`, poprzedzając ją oznaczeniem odpowiedniego podpunktu.

- a) Ile liczb parzystych jest w pliku `cyfry.txt`?
- b) Podaj liczbę z pliku `cyfry.txt`, której suma cyfr jest **największa**, oraz liczbę z tego pliku, której suma cyfr jest **najmniejsza**. W obu przypadkach jest tylko jedna taka liczba.

#### Przykład:

Dla danego zbioru liczb:

121324  
66562  
675100  
1187010

odpowiedzią są liczby: **66562** oraz **121324**, ponieważ suma cyfr liczby 66562 jest równa 25 ( $6+6+5+6+2$ ) i jest największą taką sumą, zaś suma cyfr liczby 121324 ( $1+2+1+3+2+4$ ) jest równa 13 i jest najmniejszą taką sumą.

- c) Wypisz wszystkie liczby z pliku `cyfry.txt`, których cyfry tworzą ciąg rosnący.

#### Przykład:

Cyfry liczby 123579 tworzą ciąg rosnący, ponieważ  $1 < 2 < 3 < 5 < 7 < 9$ .

Cyfry liczby 1232 nie tworzą ciągu rosnącego, ponieważ ostatnia cyfra (2) nie jest większa od przedostatniej (3).

Cyfry liczby 34556 nie tworzą ciągu rosnącego, ponieważ cyfra trzecia (5) i cyfra czwarta (5) są sobie równe.

Do oceny oddajesz plik `zadanie4.txt` oraz plik(i) .....  
tu wpisz nazwę(y) pliku (ów)  
zawierający(e) komputerową(e) realizację(e) Twojego rozwiązania.

Sprawdzane umiejętności						
W zadaniu były badane umiejętności z II i III obszaru standardów. Zdający:						
<ul style="list-style-type: none"><li>• posługuje się kompilatorem wybranego języka programowania (II.2),</li><li>• formułuje informatyczne rozwiązanie problemu przez dobór algorytmu oraz odpowiednich struktur danych i realizuje je w wybranym języku programowania (III.2).</li></ul>						
Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
4.a	pisze program sprawdzający parzystość podanych liczb.	2	0,21	0,21	–	0,21
4.b	pisze program, podający liczbę o największej i najmniejszej sumie cyfr.	4	0,07	0,11	–	0,06
4.c	pisze program, podający liczby, których cyfry tworzą ciąg rosnący.	4	0,07	0,06	–	0,07

**Przykładowy poprawny zapis rozwiązania:**

a)

odp.: liczba parzystych elementów – 495

b)

odp.: liczba o największej sumie cyfr – 187869866 (suma cyfr – 59)

liczba o najmniejszej sumie cyfr – 10010 (suma cyfr – 2).

c)

odp.: 2389, 23567, 123456789, 3468

**przykładowe rozwiązanie w C++:**

```
#include <iostream>
#include <fstream>

using namespace std;

int sumacyfr(long int n)
{
    int suma = 0;
    while(n>0)
    {
        int ost_cyfra = n%10;
        suma = suma + ost_cyfra;
        n=n/10;
    }
    return suma;
}

bool rosnacy(long int n)
{
    bool rosnie=true;
    int pom=n%10;
    n=n/10;
    while(n>0)
    {
        int ost_cyfra = n%10;
        n=n/10;
        if(pom<=ost_cyfra)
        {
            rosnie=false;
            break;
        }
        pom=ost_cyfra;
    }
    return rosnie;
}

int main()
{
    fstream plik("cyfry.txt");
    long int liczba,i,max=0,min=9999,lmax,lmin;
    i = 0;

    while(!plik.eof() )
    {
        plik>>liczba;
        if(liczba%2 == 0) i++;
        if(sumacyfr(liczba)>max)
        {
            max=sumacyfr(liczba);
            lmax=liczba;
        }
    }
}
```

```

    }
    if (sumacyfr(liczba) < min)
    {
        min = sumacyfr(liczba);
        lmin = liczba;
    }
    if (rosnacy(liczba)) cout << liczba << endl;
}
cout << "Jest: " << i << " liczb parzystych" << endl;
cout << "liczba o największej sumie cyfr: " << lmax << endl;
cout << "liczba o najmniejszej sumie cyfr: " << lmin << endl;

getchar();
cin.ignore();
return 0;
}

```

### Komentarz:

Zadanie czwarte miało charakter programistyczny. W treści wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego. W podpunkcie a) wystarczyło użyć operatora `mod`, aby sprawdzić parzystość liczb oraz licznika, który zliczałby te liczby. W podpunkcie b) należało rozbić liczbę na cyfry, a następnie zsumować cyfry liczby (funkcja `sumacyfr(n)`) i sprawdzać, czy suma jest większa od ustalonego maksimum i mniejsza od ustalonego minimum. W ostatnim podpunkcie zdający miał za zadanie znaleźć te liczby z pliku `cyfry.txt`, których cyfry tworzą ciąg rosnący. Wystarczyło tu przy rozbijaniu liczby na cyfry porównywać dwie sąsiednie liczby (funkcja `rosnacy(n)`). Podane powyżej rozwiązanie nie jest rozwiązaniem jedynym.

Zadanie to miało najwyższą frakcję opuszczeń. Jeżeli podjęto próbę wykonania zadania, to najczęściej tylko dla podpunktu a). Wystąpiły również próby rozwiązywania zadania za pomocą arkusza kalkulacyjnego (zamiast napisania programu), co było niezgodne z treścią zadania i zostały ocenione na zero punktów.

### Zadanie 5. Giełda (10 pkt)

Plik `giełda.txt` składa się z 400 wierszy, w każdym wierszu znajdują się trzy liczby całkowite oddzielone odstępami, oznaczające zmiany cen akcji firm A, B i C na giełdzie papierów wartościowych w kolejnych 400 dniach. Cena jednej akcji każdej z firm przed pierwszym dniem opisanym w pliku `giełda.txt` była równa 200 zł.

#### Przykład:

Jeśli trzy pierwsze wiersze pliku `giełda.txt` są równe:

```

0 12 -7
1 -3 4
5 2 1

```

to ceny akcji firm A, B i C w trzech kolejnych dniach są równe:

Kiedy	Cena A	Cena B	Cena C
1. dzień	$200 + 0 = 200$	$200 + 12 = 212$	$200 - 7 = 193$
2. dzień	$200 + 1 = 201$	$212 - 3 = 209$	$193 + 4 = 197$
3. dzień	$201 + 5 = 206$	$209 + 2 = 211$	$197 + 1 = 198$

Korzystając z danych zawartych w pliku `giełda.txt` oraz dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź, z wyjątkiem wykresu

w podpunkcie c), zapisz w pliku zadanie5.txt, poprzedzając ją oznaczeniem odpowiedniego podpunktu.

- Wyznacz końcową (po 400 dniach) cenę jednej akcji każdej z firm A, B i C.
- Podaj najwyższą i najniższą cenę akcji każdej z firm w czasie tych 400 dni.
- Sporządź wykres liniowy ilustrujący ceny akcji wszystkich trzech firm w kolejnych 400 dniach. Oś X Twojego wykresu powinna być osią czasu, a oś Y odpowiadać wartościom akcji. Zadbaj o czytelny opis wykresu.
- Krachem giełdowym będziemy nazywać taki dzień, w którym spadła cena akcji każdej z firm, a jednocześnie suma spadków cen jest większa niż 20 zł. Podaj liczbę krachów giełdowych oraz ceny akcji wszystkich firm w momencie ostatniego krachu.

Przykład:

Poniżej prezentujemy przykładowe ceny w dwóch kolejnych dniach.

Cena A	Cena B	Cena C
233	166	164
230	150	160

Drugiego dnia wystąpił krach, ponieważ spadła cena akcji wszystkich firm, a suma spadków cen jest równa  $3 + 16 + 4 > 20$ .

- Rekordem firmy nazywamy cenę akcji w dniu, w którym jest ona wyższa od wszystkich wcześniejszych cen akcji tej firmy. Podaj, ile razy w 400 dniach opisanych w pliku giełda.txt zmieniał się rekord firmy A.

Uwaga: Zakładamy, że przed pierwszym dniem rekord był równy 200 zł.

#### Sprawdzane umiejętności

W zadaniu były badane umiejętności z II obszaru standardów. Zdający:

- dobiera właściwy program (użytkowy lub własnoręcznie napisany) do rozwiązywanego zadania (II.6),
- posługuje się arkuszem kalkulacyjnym, wykonując obliczenia przy pomocy wbudowanych funkcji i zaprojektowanych formuł (II.1),
- posługuje się arkuszem kalkulacyjnym w celu zobrazowania graficznie informacji adekwatnie do jej charakteru.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
6.a.	wyznacza końcową cenę akcji każdej z firm.	2	0,63	0,65	–	0,63
6.b.	podaje najwyższą i najniższą cenę akcji każdej z firm.	2	0,52	0,44	–	0,56
6.c.	tworzy wykres liniowy.	2	0,68	0,63	–	0,70
6.d.	oblicza liczbę krachów giełdowych.	2	0,37	0,32	–	0,39
6.e.	oblicza liczbę rekordów firmy A.	2	0,17	0,16	–	0,18

**Przykładowy poprawny zapis rozwiązania:**

a)

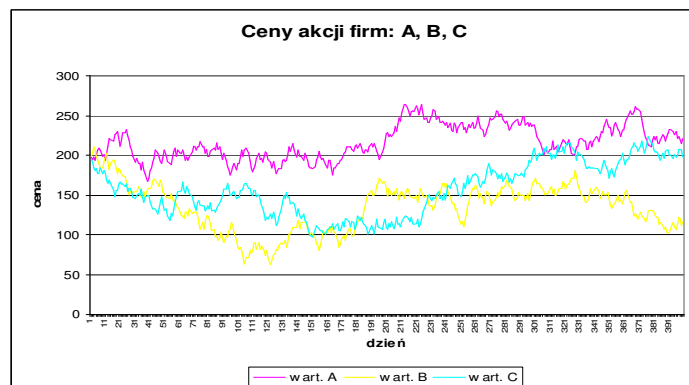
odp.: A 221, B 117, C 198

b)

odp.:

	A	B	C
max	264	210	223
min	167	62	97

c)



d)

odp.: Liczba krachów – 6

Ceny w momencie ostatniego krachu: A 224, B 163, C 194

e)

odp.: liczba rekordów – 14.

**Komentarz:**

Zadanie 5 składa się z pięciu punktów a) – e). Wszystkie dotyczą zestawu danych z tabeli opisującej zmiany cen akcji firm A, B i C na giełdzie papierów wartościowych w kolejnych 400 dniach. Naturalnym narzędziem do rozwiązania tego zadania jest arkusz kalkulacyjny. Uzyskanie odpowiedzi dla punktów a) i b) było stosunkowo proste, wymagało jedynie zastosowania prostych formuł i standardowych funkcji w odniesieniu do wierszy/kolumn tabeli.

W podpunkcie c) należało sporządzić wykres liniowy ilustrujący ceny akcji wszystkich trzech firm w kolejnych 400 dniach.

Najtrudniejsze dla zdających, choć wymagające zastosowania prostych formuł i funkcji logicznych, okazały się punkty d) i e). Aby rozwiązać pierwszy z nich, należało sprawdzić, czy wszystkie zmiany cen akcji są ujemne, jeśli tak, to należało je zsumować i sprawdzić, czy ich wartość bezwzględna jest większa od 20. W punkcie e) wystarczyło wykorzystać funkcję logiczną JEŻELI, w której testem logicznym byłoby sprawdzanie, czy aktualna wartość komórki jest wyższa od maksimum ze wszystkich poprzednich wartości akcji firmy.

**Zadanie 6. Rekrutacja (10 pkt)**

W plikach `kandydaci.txt` i `zgloszenia.txt` znajdują się informacje o naborze na studia w pewnej uczelni wyższej. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym (nie ma w nim właściwych danych). Dane w poszczególnych wierszach rozdzielone są pojedynczymi znakami odstępu.

Jeden wiersz pliku `kandydaci.txt` zawiera następujące informacje o jednej osobie, oddzielone odstępami: identyfikator osoby (`idosoby`), imię, nazwisko, matematyka,

informatyka, fizyka, językobcy, płeć. Pola matematyka, informatyka, fizyka, językobcy zawierają liczby punktów (nieujemne liczby całkowite), uzyskanych na egzaminie maturalnym na poziomie rozszerzonym (odpowiednio) z tych przedmiotów. Wartość pola płeć to k (kobieta) lub m (mężczyzna).

**Przykład:**

k007 Jan Kowalski 70 55 60 55 m  
k001 Anna Nowak 88 56 49 66 k

Jeden wiersz pliku zgłoszenia.txt zawiera informację o zgłoszeniu jednej osoby na jeden kierunek studiów. W wierszu znajdują się następujące informacje: kierunek studiów (kierunek), identyfikator kandydata (idosoby).

**Przykład:**

polonistyka k007  
informatyka k001  
matematyka k007

Liczbą punktów rekrutacyjnych kandydata na studia nazywać będziemy sumę punktów uzyskanych z matematyki, informatyki, fizyki i języka obcego.

Wykorzystując dane zawarte w tych plikach oraz dostępne narzędzia informatyczne, wykonaj poniższe polecenia. odpowiedzi do poszczególnych podpunktów zapisz w pliku zadanie6.txt, a każdą z nich poprzedź literą oznaczającą ten podpunkt.

- Podaj osoby, które uzyskały 400 punktów rekrutacyjnych. Wynik przedstaw w postaci listy zawierającej imiona i nazwiska, uporządkowanej alfabetycznie według nazwisk.
- Utwórz zestawienie, w którym podasz, ilu kandydatów zgłosiło się na informatykę, ilu na matematykę, a ilu na fizykę.
- Podaj imiona i nazwiska wszystkich tych osób, które zgłosiły się jako kandydaci dokładnie na pięć kierunków studiów.
- Podaj liczbę kobiet oraz liczbę mężczyzn, którzy zgłosili się jako kandydaci na informatykę.
- Oblicz średnią liczbę punktów z matematyki, uzyskanych przez kandydatów, którzy zgłosili się na fizykę. Wynik zaokrąglaj do dwóch miejsc po przecinku.

Do oceny oddajesz plik(i) o nazwie(ach) .....  
tu wpisz nazwę(y) pliku(ów)

zawierający(e) komputerową(e) realizację(e) Twoich obliczeń oraz plik tekstowy zadanie6.txt, zawierający odpowiedzi do podpunktów a), b), c), d) i e) zadania (odpowiedź do każdego podpunktu poprzedź literą oznaczającą ten podpunkt).

**Sprawdzane umiejętności**

W zadaniu były badane umiejętności z II i III obszaru standardów. Zdający:

- projektuje relacyjne bazy danych i wykorzystuje do ich realizacji system bazy danych (III.3),
- stosuje metody wyszukiwania i przetwarzania informacji w relacyjnych bazach danych (II.4).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
5.a.	grupuje, sumuje i filtruje dane.	2	0,59	0,58	–	0,59
5.b.	filtruje i zlicza rekordy.	2	0,63	0,60	–	0,64

5.c.	grupuje, sumuje i filtruje dane.	2	0,34	0,32	–	0,36
5.d.	grupuje, sumuje i filtruje dane.	2	0,40	0,44	–	0,39
5.e.	wykorzystuje grupowanie danych i funkcje agregujące.	2	0,31	0,40	–	0,28

**Przykładowy poprawny zapis rozwiązania:**

a)

odp.:

imie	nazwisko
Piotr	Bogasiuk
Anastazja	Chojnawiak
Marcin	Chojnik
Karol	Kaczmarczyk
Dominik	Kaczmarek
Mariusz	Lis
Katarzyna	Weber

b)

odp:

informatyka	55
matematyka	38
fizyka	48

c)

odp.: Maciej Janczak, Krzysztof Peszek

d)

odp.: kobiety – 15, mężczyźni – 40.

e)

odp.: 71,56 ( $\pm 0,01$ ).

**Komentarz:**

Zadanie 6 miało charakter bazodanowy. Dane do zadania umieszczono w dwóch plikach: `kandydaci.txt` i `zgloszenia.txt`. Pliki te odpowiadały dwóm tabelom relacyjnej bazy danych. Tabele powiązane były relacją „jeden do wielu” poprzez identyfikator osoby. Na zadanie składało się pięć podpunktów a) – e). Realizacja podpunktu a) wymagała uzyskania informacji z jednej tabeli `kandydaci`, sumowania punktów uzyskanych przez kandydatów i filtrowania danych. Udzielenie odpowiedzi na podpunkt b) wymagało wydobycia informacji z tabeli `zgloszenia` poprzez filtrowanie i zliczenie rekordów. Realizacja podpunktu c) wymagała skorzystania z dwóch tabel, zastosowania funkcji agregujących i filtrowania. Uzyskanie prawidłowej odpowiedzi na podpunkt d) wymagało grupowania i filtrowania danych z obydwu tabel. W ostatniej części zdający miał za zadanie wyłuskać informacje z obu tabel, a następnie zastosować funkcję *średnia*. Zadanie bazodanowe okazało się dla zdających umiarkowanie trudne.

# Arkusz I

### Zadanie 1. Funkcja rekurencyjna (8 pkt)

Dana jest liczba naturalna  $n > 0$  i tablica różnych liczb całkowitych  $a[1..n]$ . Rozważamy następującą **rekurencyjną** funkcję  $F$  z argumentem  $i$ , będącym liczbą naturalną,  $1 \leq i \leq n$ .

### Funkcja $F(i)$

```

jeżeli  $i = n$  to
    wynikiem jest  $n$ 
w przeciwnym razie
     $j := F(i + 1)$ 
    jeżeli  $a[i] < a[j]$  wtedy
        wynikiem jest  $i$ 
    w przeciwnym razie
        wynikiem jest  $j$ 

```

- a) Dla danej 10-elementowej tablicy  $a = [5, 1, 8, 9, 7, 2, 3, 11, 20, 15]$  podaj w poniższej tabeli wynik wywołania funkcji  $F$  dla danego argumentu  $i$ .

$i$	$F(i)$
9	
7	
5	

A large grid of 20 columns and 10 rows, intended for drawing a picture. The grid is composed of small squares, with a slightly larger square at the top left corner.

- b) Niech  $w$  będzie wynikiem wywołania funkcji  $F$  dla argumentu  $i$ ,  $1 \leq i \leq n$ . Wtedy  $a[w]$  w odniesieniu do pozostałych liczb w tablicy  $a$  jest zawsze
- najmniejszą liczbą w tej tablicy.
  - najmniejszą liczbą w tej tablicy spośród elementów o indeksach od  $i$  do  $n$ .
  - najmniejszą liczbą w tej tablicy spośród elementów o indeksach od 1 do  $i$ .
- Podkreśl właściwą odpowiedź.





**Przykładowy poprawny zapis rozwiązania:**

a)

$i$	$F(i)$
9	10
7	7
5	6

b)

odp: najmniejszą liczbą w tej tablicy spośród elementów o indeksach od  $i$  do  $n$ .

c)

odp.: 1500

d)

Przykładowy iteracyjny zapis funkcji  $F$ :**Funkcja**  $F(i)$ 

1.  $ind\_min := n$
2. *jeśli*  $i = n$  *przejdź do kroku 4*
3. *od*  $k = n-1$  *do*  $i$   
*jeśli*  $a[k] < a[ind\_min]$  *to*  $ind\_min := k$   
*wypisz*  $ind\_min$

**Komentarz:**

Zadanie pierwsze podzielone zostało na cztery podpunkty. Dwa pierwsze z nich sprawdzały zrozumienie rekurencji, techniki kluczowej w algorytmice, choć niełatwej koncepcyjnie. Zadaniem maturzysty było wykazanie się zrozumieniem działania podanej w treści zadania funkcji rekurencyjnej. W pierwszym punkcie należało podać wynik wywołania funkcji  $F$  dla wskazanego argumentu. Wynikiem działania funkcji jest indeks najmniejszego elementu tablicy  $a[1..n]$  z zakresu od  $n$  do  $i$ , gdzie  $i$  jest parametrem funkcji. Punkt b) jest kontynuacją analizy algorytmu z punktu a), a zdający miał w pośredni sposób podać wynik działania funkcji  $F$ . W punkcie c) należało natomiast wyznaczyć liczbę porównań wykonywanych przez funkcję  $F$  dla  $i=512$  oraz tablicy składającej się z 2012 elementów. Ostatni punkt (o umiarkowanej trudności) wymagał samodzielnego skonstruowania i zapisu algorytmu iteracyjnego, którego wynikiem jest indeks minimum z zakresu od  $i$  do  $n$ . Pojawiające się błędne odpowiedzi polegały na wypisywaniu wartości, a nie indeksu minimum lub na rekurencyjnym zapisie algorytmu.

Liczbę naturalną  $n$  będziemy nazywać liczbą **osiągalną**, jeżeli istnieje takie  $k$ , że  $n = k + s(k)$ , gdzie  $k$  jest liczbą naturalną, a  $s(k)$  jest sumą cyfr liczby  $k$  w zapisie dziesiętnym.

Liczba $n$	$k$	$s(k)$	Czy $n$ jest osiągalna?
505	491	14	TAK
20	—	—	NIE
28			
31			

[illegible]

Wynik: liczba  $k$  taka, że  $n = k + s(k)$ , gdy liczba  $n$  jest osiągalna; komunikat NIE, gdy  $n$  nie jest osiągalna

[illegible]

**Sprawdzane umiejętności**

W zadaniu były badane umiejętności z I i II obszaru standardów. Zdający:

- zna techniki algorytmiczne i algorytmy (własności liczb całkowitych i naturalnych) (I.7),
- analizuje liczbę wykonywanych operacji w algorytmie (I.9),
- stosuje kolejne etapy prowadzące do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania (II.2).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
2.a	sprawdza własności liczb naturalnych.	2	0,91	0,90	–	0,93
2.b	zauważa i uzasadnia zależności matematyczne.	1	0,94	0,96	–	0,90
2.c	tworzy algorytm sprawdzający własności liczby.	4	0,60	0,64	–	0,54

**Poprawny zapis rozwiązania:**

a)

odp: dla  $n=28$ ,  $k=23$ ,  $s(k)=5$ , TAK  
dla  $n=31$  liczba nieosiągalna, NIE

b)

odp.: Ponieważ suma cyfr liczby trzycyfrowej lub czterocyfrowej nie może być większa od 36.

c)

Przykładowy algorytm:

$i:=1$

$k:=n$

dopóki  $i \leq 36$  wykonuj

$k:=n-i$

$s:=(k \text{ div } 1000) + ((k \text{ mod } 1000) \text{ div } 100) +$   
 $((k \text{ mod } 100) \text{ div } 10) + (k \text{ mod } 10)$

jeżeli  $k+s=n$  to

wypisz  $k$  i zakończ algorytm

w przeciwnym razie

$i:=i+1$

wypisz NIE

**Komentarz:**

W zadaniu drugim zdefiniowano tzw. liczby osiągalne. Zdający miał za zadanie sprawdzić, czy podane dwie liczby są osiągalne, i jeżeli tak, to podać składniki sumy  $k$  i  $s(k)$ . Kolejno należało uzasadnić zależność matematyczną. Obydwa punkty okazały się dla maturzystów łatwe, co świadczy o dobrej podbudowie umiejętności matematycznych, reprezentowanych przez zdających. W punkcie c) należało samodzielnie zapisać optymalny algorytm sprawdzający, czy liczba naturalna  $n$  z przedziału  $[1000, 9999]$  jest liczbą osiągalną, i jeśli tak, to podać składową  $k$ . Najczęściej pojawiające się błędy polegały na nieprawidłowym zapisie warunków pętli, niewykorzystywaniu informacji z punktu b), czyli sprawdzaniu  $k$  od 1 zamiast od  $n-36$  oraz wypisywaniu (zamiast liczby  $k$ ) komunikatu TAK, NIE.

**Zadanie 3. Test (5 pkt)**

Podpunkty a) – e) zawierają po cztery odpowiedzi. Zdecyduj, które z podanych odpowiedzi są prawdziwe (P), a które fałszywe (F). **Zaznacz znakiem X** odpowiednią komórkę w tabeli. W każdym pytaniu uzyskasz punkt tylko za komplet poprawnych odpowiedzi.

- a) Poniżej przedstawiono fragment bazy danych zawierającej informacje o książkach, czytelnikach i wypożyczeniach. Pole *id\_Cz* w tabeli *Czytelnicy* jest połączone relacją „jeden do wielu” z polem *id\_Cz* w tabeli *Wypożyczenia*, podobnie pole *id\_Ks* w tabeli *Książki* z polem *id\_Ks* w tabeli *Wypożyczenia*.

**Książki**

id_Ks	Autor	Tytuł	Rok wydania
1	John Tolkien	Hobbit, czyli tam i z powrotem	2007
2	Ursula K. Le Guin	Czarnoksiężnik z Archipelagu	2009
3	Peter V. Brett	Malowany człowiek. Księga II	2011
4	Stanisław Lem	Bajki robotów	2006
5	Trudi Canavan	Misja Ambasadora	2011
6	John Tolkien	Dzieci Hurina	2010
7	Andrzej Sapkowski	Krew Elfów	2010

**Czytelnicy**

id_Cz	Imie	Nazwisko	Klasa
1	Anna	Tulik	I
2	Magda	Nowak	I
3	Marek	Krokus	I
4	Jacek	Doniec	II
5	Wojtek	Madejski	II
6	Michał	Sośnierz	II
7	Franek	Jedliński	II
8	Sandra	Biecz	III
9	Jowita	Kolska	III
10	Ala	Mleczko	III

**Wypożyczenia**

Nr_W	id_Ks	id_Cz
1	4	2
2	7	3
3	3	9
4	1	5
5	2	8
6	4	10
7	6	8
8	5	7
9	5	9
10	7	1

Z danych umieszczonych w tym fragmencie bazy wynika, że

	P	F
Jowita Kolska wypożyczyła „Misję Ambasadora”.		
„Bajki Robotów” były wypożyczane dwa razy.		
Z podanych klas (I, II, III) najwięcej książek wypożyczyli uczniowie klasy II.		
Jacek Doniec nie wypożyczył jeszcze żadnej książki.		

b) Które z podanych pojęć i skrótów dotyczą technologii i standardów wykorzystywanych w budowie lokalnych sieci komputerowych?

	P	F
XML		
WiFi		
Ethernet		
Telnet		

c) Liczba  $1E_{(16)}$  jest równa liczbie

	P	F
$101010_{(2)}$ .		
$36_{(8)}$ .		
$1110_{(3)}$ .		
$30_{(10)}$ .		

d) Dla dwóch liczb  $1110_{(2)}$  i  $10_{(2)}$ , ich

	P	F
suma jest równa $10000_{(2)}$ .		
różnica jest równa $1000_{(2)}$ .		
iloczyn jest równy $11110_{(2)}$ .		
iloraz jest równy $111_{(2)}$ .		

e) Licencja GNU GPL zezwala na

	P	F
uruchamianie programu do użytku domowego.		
rozpowszechnianie niezmodyfikowanej kopii programu.		
analizowanie, jak program działa i dostosowywanie go do swoich potrzeb.		
udoskonalanie programu i publiczne rozpowszechnianie własnych ulepszeń.		

**Sprawdzane umiejętności**

W zadaniu były badane umiejętności z I obszaru standardów. Zdający:

- analizuje problem i zbiór danych (III.3),
- zna podstawową terminologię związaną z sieciami komputerowymi: rodzaje sieci, protokoły (I.4),
- zna systemy liczbowe mające zastosowanie w informatyce (I.3),
- zna systemy liczbowe mające zastosowanie w informatyce (I.3),
- zna i opisuje zasady etyczne i prawne związanych z wykorzystywaniem informacji i oprogramowania (I.11).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a	wyszukuje informacje relacyjnej bazy danych.	1	0,92	0,94	–	0,90
3.b.	rozróżnia pojęcia dotyczące technologii i standardów wykorzystywanych w budowie lokalnych sieci komputerowych.	1	0,70	0,71	–	0,68
3.c.	zapisuje liczby w różnych systemach liczbowych.	1	0,74	0,71	–	0,78
3.d.	wykonuje podstawowe działania na liczbach binarnych.	1	0,82	0,79	–	0,88
3.e.	Zna zasady korzystania z oprogramowania z licencją GNU GPL.	1	0,45	0,39	–	0,56

**Poprawny zapis rozwiązania:**

a	PPFP
b	FPPF
c	FPFP
d	PFFP
e	PPPP

**Komentarz:**

Zadanie miało charakter testu wyboru, sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Składało się 5 pytań a) – e) wielokrotnego wyboru (wybór z czterech odpowiedzi). Pytanie a) dotyczyło znajomości zasad budowy i funkcjonowania relacyjnej bazy danych. Punkt c) dotyczył pojęć związanych z sieciami komputerowymi. Punkty c) i d) wymagały znajomości metod zamiany między pozycyjnymi systemami liczbowymi i podstawowych operacji matematycznych w systemie binarnym. Na koniec, w punkcie e), sprawdzana była wiedza na temat praw użytkownika korzystającego z oprogramowania z licencją GNU GPL. Zadanie okazało się łatwe, zdający wykazali się dobrą znajomością poruszanych w zadaniu zagadnień.

## Arkusz II

### Zadanie 4. Szyfr (10 pkt)

Rozważmy szyfr podstawieniowy, działający zgodnie z następującymi zasadami:

- Tekst jawny, szyfrogram oraz klucz składają się wyłącznie z wielkich liter alfabetu angielskiego.
- Litery ponumerowano i przyporządkowano im kody ASCII (liczby z zakresu 65–90):

Tabela numerów i kodów ASCII poszczególnych liter

Litera	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
Nr litery	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	26
Kod ASCII	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

- Kolejne litery tekstu jawnego są szyfrowane za pomocą kolejnych liter słowa będącego kluczem, być może powtórnego wiele razy.
- W procesie szyfrowania tekst jawny przekształcany jest na szyfrogram przy pomocy klucza poprzez dodanie do **kodu** litery tekstu jawnego **numeru** odpowiadającej jej litery klucza. Jeżeli tak uzyskana wartość liczbową będzie większa od 90, należy ją zmniejszyć o 26. Szyfrem danej litery jest litera o tak uzyskanym kodzie. Poniższy przykład precyzuje zasady szyfrowania.

#### Przykład:

*Tekst jawny: LATO, klucz: WODA*

$L+W = 76+23 = 99$ . Ponieważ przekroczono zakres 90, należy od 99 odjąć 26, czyli  $99-26 = 73$ . Zatem zaszyfrowanym znakiem jest litera I.

$A+O = 65+15 = 80$ , czyli zaszyfrowanym znakiem jest litera P.

$T+D = 84+4 = 88$ , czyli zaszyfrowanym znakiem jest litera X.

$O+A = 79+1 = 80$ , czyli zaszyfrowanym znakiem jest litera P.

*Szyfrogram: IPXP*

- Jeżeli użyte słowo kluczowe jest zbyt krótkie, by wystarczyło do zaszyfrowania całego tekstu, należy użyć jego powtórzeń.

#### Przykład:

*Tekst jawny: MARTA, klucz: TOR*

$M+T = 77+20 = 97$ ,  $97-26=71$ , G

$A+O = 65+15 = 80$ , P

$R+R = 82+18 = 100$ ,  $100-26 = 74$ , J

$T+T = 84+20 = 104$ ,  $104-26=78$ , N

$A+O = 65+15 = 80$ , P

*Szyfrogram: GPJNP*

- W procesie deszyfrowania szyfrogram powinien być przekształcany na tekst jawny przy pomocy klucza poprzez odjęcie od kodu litery szyfrogramu numeru odpowiadającej jej litery klucza (jeżeli tak uzyskana wartość liczbową będzie mniejsza od 65, należy ją powiększyć o 26), a następnie powinna być odczytana litera o otrzymanym kodzie.



Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia.

- a) W pliku `tj.txt` znajdują się niezaszyfrowane słowa, a w pliku `klucze1.txt` – klucze szyfrujące. W obu plikach wyrazy umieszczone są w osobnych wierszach. Zaszifruj słowa zawarte w pliku `tj.txt`, wynik zapisz w pliku `wynik4a.txt`. Wyraz zapisany w  $N$ -tym wierszu w pliku z wynikami powinien stanowić szyfrogram tekstu jawnego, znajdującego się w  $N$ -tym wierszu w pliku z tekstem jawnym, uzyskanym za pomocą klucza znajdującego się w  $N$ -tym wierszu pliku z kluczami.
- b) W pliku `sz.txt` znajdują się zaszyfrowane słowa, a w pliku `klucze2.txt` znajdują się klucze deszyfrujące. W obu plikach wyrazy umieszczone są w osobnych wierszach. Odszyfruj słowa zawarte w pliku `sz.txt`, wynik zapisz do pliku `wynik4b.txt`. Wyraz zapisany w  $N$ -tym wierszu w pliku z wynikami powinien stanowić tekst jawny szyfrogramu znajdującego się w  $N$ -tym wierszu w pliku z szyfrogramami, uzyskanymi za pomocą klucza zapisanego w  $N$ -tym wierszu pliku z kluczami.

Do oceny oddajesz plik(i) o nazwie(ach) .....  
tu wpisz nazwę(y) pliku(ów)  
 zawierający(e) komputerową(e) realizację(e) poleceń oraz pliki tekstowe: `wynik4a.txt` i `wynik4b.txt` zawierające wyniki do podpunktów a) i b).

### Sprawdzane umiejętności

W zadaniu były badane umiejętności z III obszaru standardów. Zdający:

- zna techniki algorytmiczne i algorytmy operujące na tekstach, stosuje je w szyfrowaniu podstawieniowym (I.7),
- formułuje informatyczne rozwiązanie problemu przez dobór algorytmu i odpowiednich typów oraz struktur danych i zaimplementowanie go w wybranym języku programowania (szyfrowanie z zastosowaniem klucza) (III.2).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.	Szyfruje słowa za pomocą słowa – klucza.	6	0,45	0,47	–	0,40
2.	Deszyfruje za pomocą klucza deszyfrującego.	4	0,41	0,45	–	0,35

### Przykładowy poprawny zapis rozwiązania:

a) odp.:

CNA UUU WNS TRX WCCG YOYG XEUUAU WVBY QIE FYDBZ YQRQXDMHHAOS RKXETREBHGFBW WYAUKDWWGTI LWTZMEPL MLWGODX GEWPZWCR XGZMRSAD GZCPVLHDUF GDDULBERQA	SFNBNAQSKV MPEYAOA MWLPBGPF LNPFMBNPPDOR APPSHCYCU TUNMIVNB PFRNMUYDHOY WKNLJPTWNHURN QDYLEQFMTUHNH GYIGEBTKTDVD VHARCCKPDFYLQ WSQILBTJJD TDLKXPEGAN GZIHOMZMTAXI EZJTBGIQ XJZSJSVXPM AWNJXL0MVNNX WFD0GBEAGSRGL ZKJSF0SRNYGJ	OHCOZZWENH PEOEGGZLBYPNV JJJOUTTRCCXPVJ IFPIGOAAXE MFRAMXPFEJY ARJHVBEJUL SIDIAXMPVS IPGBQRUJU XRTXMFKSFBOT LNBUIAPOB MPKRROJMCUMXFLVYN PCVWLKZPUJP GFOQXCWFMKB TVJZKBVO MWWCLSCIE TPJZIKVSBZEDPRCZC EDNXUYBVYNKNN WVZSFSUHBNN YYFUHXAIZL
---	---	---

APEUDKNPMO GMFCJJFYTMH PPPKMWIRAU LFXXPRJZL OYYZCKVUY XKMQNSG NZSTFEGHVCE ONRVWVOW MMMICLXVQ MRTCGISKFY SNYN SHBP EHDABER KYPGTMDUF NIYBOQJ EQQRIDJL ADWPDRQZL QAYMTFYSM QJJUCQO OPPICXNLP XJFHQBKBSX KHZVSSXRD	YEVHEITYIZOU ELZBZOYOL CSQRXMORC SKFOMNFAHLEQXD NQJFXSUAGEZ WBVYTBVU SFDKNNMPJIBAOB AFOJNJXXEL DUHSRGIHLPJHG DRLIPMGSV OYYSFWYSHZ NKYCOKJDG FAVAJFNXEZB BUUFZWJQJORHSK GTGPMTYPMBZN IAYGSDHKFNP TAIUQSODRLIE GOWCZFXJFENJQ VPIDTXAOCCOB CPJPTGEBMZ JHUYVFCEASE	DHHABOENDL TBLHVADDPHC WJYSAEUAYCG DKMXUKTNAQRC BVKIQVFSHETL SHSMRDFUGD COVYNVPDX FTKOHQIFNNU XQMBOHIPYDAMO IRGRCVPIBFUEWZYQR YSQWBXFQZRCB DLEDES GYEAF IELOZ VIKYTW NKQ OXp ALX ZQHR NSOZ MDO
---	--	--

```

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main()
{
    ifstream wejście("tj.txt");
    ifstream wejście2("klucz.txt");
    ofstream wyjście("wynik4a.txt");

    string słowo, klucz;

    while(!wejście.eof())
    {
        wejście>>słowo;
        wejście2>>klucz;
        for(int i=0; i<słowo.length(); i++)
        {
            if(i>klucz.length()-1)
            {
                if(int(słowo[i])+int(klucz[i-klucz.length()]-64)>90)
                {
                    wyjście<<char(int(słowo[i])+int(klucz[i-klucz.length()]-64)-
26);
                }
                else
                {
                    wyjście<<char(int(słowo[i])+int(klucz[i-klucz.length()-64]));
                }
            }
            else if((int(słowo[i])+int(klucz[i])-64)>90)
            {
                wyjście<<char(int(słowo[i])+int(klucz[i]-64)-26);
            }
            else
            {
                wyjście<<char(int(słowo[i])+int(klucz[i])-64);
            }
        }
    }

```

```

        wyjscie<<endl;;
    }
    wejscie.close();
    wejscie2.close();
    wyjscie.close();

    cin.ignore();
    getchar();
    return 0;
}

```

b)

odp.:

KOT PIES COS NIC NUTA BUT LUPA ZUPA KWIAT CEL KONWERTEREM ZAAKCEPTOWANIA KOLOROWANIE PIERWOTNE KLASYCZNY STOPOWYCH PNEUMATYCZNE PODSTAWOWYCH ROZPOZNAJE POCHODZENIA WYSZUKIWANIE PRZEKAZYWANY PRZYDATNE AUTOMATYCZNE PRZYGOTOWYWANIU PRZERZUTKA ROZPOZNAWANIE PROGRAMU ZAPISYWANE DOPASOWYWANIE REJESTRACJI TURBULENCJE KODOWANIA ZINTEGROWANE NAJPOPULARNIEJSZE MECHANICZNE HYDRAULICZNE PRZEDNIEGO ZASTOSOWANIE TEKSTOWYCH	NATURALNEJ WYSZUKIWANIU KOMPUTERACH JEDNOCZESNYM KONFIGUROWALNY KOMENTARZACH ZACHOWANIEM ZAOPATRZONY ROZPISANEGO SAMOCHODU RZECZYWISTY ELEKTRYCZNY OBUDOWIE GUBERNATOR KONWERTUJE ZAAWANSOWANYCH KONSTRUKTORA POWIERZCHNIOWYCH ZLIKWIDOWANIE HAMOWANIA PUNKTOWYM SZKIELETEM SAMOLOTU POPULARNYCH CESARZOWI PROGRAMOWANIA UDZIELANIE TABULACJI POMOCNICZYCH KOMPONENTY STANDARDY REGULARNYCH KLAWISZOWY UPRAWNIENIAMI DOWOLNYCH WSTAWIANIA USTAWIENIA NAZWISKU NIEZNACZNIE ORGANIZACJI	POZOSTAJE DODATKOWYCH ZAPEWNIENIE ZAKTUALIZOWANO ZAZWYCZAJ LECZNICZEJ STOSOWANIA TOWARZYSZY PODOBNYM ZAJMOWANEJ ZASTOSOWANIA POKRYWANIA SYSTEMOWEGO BIODROWYM WSZYSTKICH POWIERZCHNI DEKORACYJNEGO ORGANIZACJA POTRANATEM ZAMIENNIKIEM WIZYJNE NAJLEPIEJ DOCELOWYM KONFRONTACJI ORGANIZOWANE OCHRONNEGO INSTALACJI DOSTOSOWANIE NINIEJSZEJ WSZYSTKIE KREM KTOS ZNAC ALBO RYBA LUB MUS PLUS KROWA OWCA
---	--	--

```

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main()

```

```

{
ifstream wejscie("sz.txt");
ifstream wejscie2("klucze2.txt");
ofstream wyjscie("wynik4b.txt");

string slowo, klucz;

while(!wejscie.eof())
{
    wejscie>>slowo;
    wejscie2>>klucz;
    for(int i=0; i<slowo.length(); i++)

        if(i>klucz.length()-1)
        {
            if(int(slowo[i])-int(klucz[i-klucz.length()])+64<65)
            {
                wyjscie<<char(int(slowo[i])-int(klucz[i-
klucz.length()]+64)+26);
            }
            else
            {
                wyjscie<<char(int(slowo[i])-int(klucz[i-klucz.length()]+64));
            }
        }
        else if((int(slowo[i])-int(klucz[i])+64)<65)
        {
            wyjscie<<char(int(slowo[i])-int(klucz[i]+64)+26);
        }
        else
        {
            wyjscie<<char(int(slowo[i])-int(klucz[i])+64);
        }
    }
    wyjscie<<endl;;
}

wejscie.close();
wejscie2.close();
wyjscie.close();

cin.ignore();
getchar();
return 0;
}

```

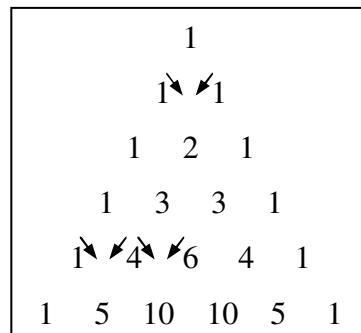
#### **Komentarz:**

Zadanie 4 miało charakter programistyczny. Wyróżniało się tym, iż w treści nie wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego, ale wybór innego narzędzia skazywał zdającego na niepowodzenie. Algorytmicznie proste zadanie należało zaimplementować w wybranym języku programowania. W podpunkcie a) zdający otrzymał plik tekstowy `tj.txt` składający się z 120 słów, które należało zaszyfrować za pomocą słów kluczy zapisanych w pliku `kuczel.txt`. Metoda szyfrowania została dokładnie opisana w treści zadania i zobrazowana przykładami. Pierwsze 10 słów i ostatnie 10 słów w pliku ma taką samą długość, jak długość odpowiadającego mu klucza. Pozostałe słowa jawne i odpowiadające im klucze są różnej długości. Taki układ pozwalał wyłuskać błędy w warunkach pętli i odpowiednio ocenić rozwiązanie. W podpunkcie b) tego zadania maturzysta miał za zadanie odszyfrować zakodowane słowa w pliku `sz.txt` przy pomocy słów kluczy zawartych

w pliku `klucze2.txt`. Wskaźniki łatwości dla tego zadania klasyfikują je jako zadanie umiarkowanie trudne.

### Zadanie 5. Trójkąt Pascala (10 pkt)

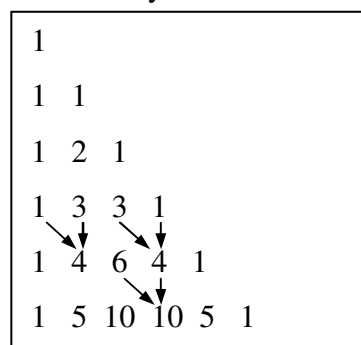
Trójkąt Pascala to trójkątna tablica liczb, skonstruowana w następujący sposób: na dwóch bokach trójkąta znajdują się liczby 1, kolejne liczby wewnątrz trójkąta obliczane są poprzez zsumowanie dwóch najbliższych liczb położonych w wierszu powyżej (rysunek 1a).



Rysunek 1a

Tablica liczb może przyjąć postać trójkąta prostokątnego, w której jedynkami wypełniona jest przyprostokątna pionowa i przeciwprostokątna (rysunek 1b).

Obliczanie pozostałych liczb wykonuje się na tej samej zasadzie jak podano powyżej, tzn. poprzez zsumowanie dwóch liczb położonych w wierszu powyżej – jednej, znajdującej się nad obliczaną sumą, i drugiej, położonej na lewo od pierwszego składnika sumy.



Rysunek 1b

Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. odpowiedzi do podpunktów a), b), i c) zapisz w pliku `wynik5.txt`, a każdą z nich poprzedź literą oznaczającą ten podpunkt.

Poniższe polecenia odnoszą się do trójkąta Pascala, składającego się z 30 wierszy. Wiersze są numerowane od 1.

- Podaj największą liczbę spośród liczb wchodzących w skład 10-tego, 20-tego i 30-tego wiersza trójkąta Pascala.
- Utwórz zestawienie zawierające dla każdego wiersza trójkąta Pascala: jego numer oraz liczbę cyfr (nie liczb) znajdujących się w tym wierszu.
- Podaj numery wierszy, które nie zawierają liczb podzielnych przez 5.
- Występowanie liczb parzystych i nieparzystych w trójkącie Pascala układa się we wzór przypominający klasyczny fraktal nazywany „trójkątem Sierpińskiego” (rysunek 2).



Rysunek 2

W oparciu o zbudowany trójkąt Pascala (prostokątny lub równoramienny), utwórz graficzny rozkład liczb znajdujących się w trójkącie, które są podzielne przez 3, np. oznaczając przez „X” komórki z liczbami podzielnymi przez 3 lub wypełniając je czarnym kolorem tła, zaś zawartości każdej z pozostałych komórek oznacz znakami odstępu (spacji) i białym kolorem tła.

Do oceny oddajesz plik(i) o nazwie(ach) .....  
tu wpisz nazwę(y) pliku (ów)  
 komputerową(e) realizację(e) Twoich obliczeń, plik tekstowy `wynik5.txt` zawierający odpowiedzi do podpunktów a), b) i c) oraz plik .....  
tu wpisz nazwę pliku  
 zawierający reprezentację graficzną rozwiązania podpunktu d) zadania.

### Sprawdzane umiejętności

W zadaniu były badane umiejętności z II i III obszaru standardów. Zdający:

- stosuje odpowiednie formatowanie danych i tabeli oraz wykonuje obliczenia przy pomocy wbudowanych oraz zaprojektowanych formuł (II.1),
- projektuje i przeprowadza wszystkie etapy na drodze do otrzymania informatycznego rozwiązania problemu (III.2),
- tworzy dokument graficzny (III.4).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
5.a.	konstruuje trójkąt Pascala i wyszukuje maksymalne liczby w odpowiednich wierszach.	3	0,73	0,77	–	0,66
5.b.	oblicza liczbę cyfr w wierszu.	2	0,43	0,48	–	0,37
5.c.	sprawdza podzielność liczb przez 5.	2	0,59	0,66	–	0,47
5.d.	tworzy graficzny rozkład liczb, które są podzielne przez 3.	3	0,49	0,60	–	0,32

wiersz 10: 126

wiersz 30: 77 558 760

1. 1	11. 25	21. 88
2. 2	12. 28	22. 96
3. 3	13. 31	23. 102
4. 4	14. 38	24. 114
5. 5	15. 44	25. 125
6. 8	16. 50	26. 134
7. 10	17. 57	27. 142
8. 12	18. 64	28. 154
9. 14	19. 69	29. 166
10. 18	20. 76	30. 178

1,2,3,4,5,10,15,20,25

[illegible]

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int tab[30][30]; //tablica zadeklarowana globalnie wypelniona jest zerami
ofstream wyjście("wynik5.txt");

int main()
{
    int max, liczba, licznik=0;
    for(int i=0; i<30; i++)
    {
        tab[i][0]=1;
    }

    tab[1][1]=1;
    for(int i=2; i<30; i++)
```

```

{
    for(int j=1; j<=i; j++)
    {
        tab[i][j]=tab[i-1][j-1]+tab[i-1][j];
    }
}

/*for(int i=0; i<30; i++)    //trójkąt na monitorze
{
    for(int j=0; j<30; j++)
    {
        if (tab[i][j]!=0)
        {
            cout<<tab[i][j]<<" ";

        }
    }
    cout<<endl;
}*/

//podpunkt a dla 10 wiersza
max=1;
for(int i=0; i<10; i++)
{
    if(tab[9][i]>max)
    {
        max=tab[9][i];
    }
}
wyjście<<"a) " <<endl<<"największa liczba w 10 wierszu to " <<max<<endl;

//dla 20 wiersza
max=1;
for(int i=0; i<20; i++)
{
    if(tab[19][i]>max)
    {
        max=tab[19][i];
    }
}

wyjście<<"największa liczba w 20 wierszu to " <<max<<endl;
//dla 30 wiersza
max=1;
for(int i=0; i<30; i++)
{
    if(tab[29][i]>max)
    {
        max=tab[29][i];
    }
}

wyjście<<"największa liczba w 30 wierszu to " <<max<<endl<<endl<<"b)" <<endl;
//podpunkt b
for(int i=0; i<30; i++)
{
    for(int j=0; j<=i; j++)
    {
        liczba=tab[i][j];

        while(liczba!=0)
        {

```



```

        liczba=liczba/10;
        licznik=licznik+1;
    }
}
wyjście<<i+1<<"<<licznik<<endl;
licznik=0;
}

//podpunkt c
wyjście<<endl<<endl<<"c)"<<endl;
licznik=0;
for(int i=0; i<30; i++)
{
    for(int j=0; j<=i; j++)
    {
        if(tab[i][j]%5==0)
        {
            licznik=1;
            break; //licznik sluzzy w roli zmiennej logicznej
        }
    }

    if(licznik==0)
    {
        wyjście<<i+1<<endl;
    }
    else
    {
        licznik=0;
    }
}
cin.ignore();
getchar();
return 0;
}

```

#### 5.d)

```

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int tab[30][30]; //tablica zadeklarowana globalnie wypelniona jest zerami
ofstream wyjście("trojkat.txt");

int main()
{
    int max, liczba, licznik=0;

    for(int i=0; i<30; i++)
    {
        tab[i][0]=1;
    }
    tab[1][1]=1;

    for(int i=2; i<30; i++)
    {
        for(int j=1; j<=i; j++)

```

```

    {
        tab[i][j]=tab[i-1][j-1]+tab[i-1][j];
    }
}

for(int i=0; i<30; i++)
{
    for(int j=0; j<=i; j++)
    {
        if((tab[i][j]%3)==0)
        {
            wyjście<<"X";
        }
        else
        {
            wyjście<<" ";
        }
    }
    wyjście<<endl;
}
wyjście.close();

cin.ignore();
getchar();
return 0;
}

```

## Rozwiązanie w arkuszu

5.a)

zadanie5.xlsx - Mi...

Narzędzia: Wstawianie Układanie Formuły Dane Recepcja Widok

Wklej Czcionka Wyrównanie Liczba Style Komórki

Schowek

G12  $f_x = G11 + F11$

	A	B	C	D	E	F	G	H	I
2	1								
3	1	1							
4	1	2	1						
5	1	3	3	1					
6	1	4	6	4	1				
7	1	5	10	10	5	1			
8	1	6	15	20	15	6	1		
9	1	7	21	35	35	21	7	1	
10	1	8	28	56	70	56	28	8	1
11	1	9	36	84	126	126	84	36	9
12	1	10	45	120	210	252	210	120	45
13	1	11	55	165	330	462	462	330	165
14	1	12	66	220	495	792	924	792	495

Gotowy 100%

Następną czynnością jest skorzystanie z funkcji MAX, której argumentami są liczby trójkąta Pascala z wiersza 10, następnie z wiersza 20 i kolejno z wiersza 30.

5.b)

zadanie5.xlsx - Microsoft Excel

Narzędzia: Wstawianie Układanie s Formuły Dane Recenzja Widok

Wklej Czcionka Wyrównanie Liczba Style Komórki

Schowek Edycja

F43  $f_x = DŁ(TEKST(F13;"#"))$

	A	B	C	D	E	F	G	H	I	J	K
32	1	0	0	0	0	0	0	0	0	0	0
33	1	1	0	0	0	0	0	0	0	0	0
34	1	1	1	0	0	0	0	0	0	0	0
35	1	1	1	1	0	0	0	0	0	0	0
36	1	1	1	1	1	0	0	0	0	0	0
37	1	1	2	2	1	1	0	0	0	0	0
38	1	1	2	2	2	1	1	0	0	0	0
39	1	1	2	2	2	2	1	1	0	0	0
40	1	1	2	2	2	2	2	1	1	0	0
41	1	1	2	2	3	3	2	2	1	1	0
42	1	2	2	3	3	3	3	3	2	2	1
43	1	2	2	3	3	3	3	3	3	2	2

Gotowy 100%

Po obliczeniu liczby cyfr w kolejnych elementach trójkąta należy zsumować dla każdego wiersza liczby cyfr i wypisać zestawienie.

5.c)

W podpunkcie należy skorzystać z funkcji MOD w celu sprawdzenia podzielności liczb trójkąta przez 5.

5.d)

Aby utworzyć reprezentację graficzną liczb podzielnych przez 3 w trójkącie Pascala, można zastosować funkcję JEŻELI, w teście logicznym wstawić funkcję MOD, zaś jako drugi argument znak „X”. Można również zastosować formatowanie warunkowe.

**Komentarz:**

Kluczowym krokiem w kierunku rozwiązania zadania 5 jest skonstruowanie trójkąta Pascala (zasada jego tworzenia została przypomniana w treści zadania). Zadanie można rozwiązać, pisząc odpowiedni program lub stosując formuły i funkcje (matematyczne, statystyczne i logiczne) arkusza kalkulacyjnego. W ostatniej części należało sporządzić reprezentację graficzną liczb budujących trójkąt Pascala i podzielnych przez 3. Co ciekawe, pojawiały się tu błędy polegające na utworzeniu „rewersu” graficznego rozkładu liczb, czyli rozkład liczb niepodzielnych przez 3 lub błędy polegające na braku w reprezentacji graficznej kilku końcowych wierszy. Wskaźnik łatwości klasyfikuje to zadanie pośród zadań umiarkowanie trudnych.

**Zadanie 6. Tablice (10 pkt)**

Firma GoldCar wynajmuje lub oddaje w leasing pojazdy firmom na terenie całego kraju.

Dane są trzy pliki tekstowe o nazwach: `tablice.txt`; `uslugi.txt`; `nip_firm.txt`. Zawierają one informacje na temat tablic rejestracyjnych pojazdów oraz firm korzystających z usług GoldCar. Pierwszy wiersz każdego z plików jest wierszem nagłówkowym, a dane w wierszach rozdzielone są znakami tabulacji.

Plik o nazwie `tablice.txt` zawiera w każdym wierszu: *oznaczenie literowe powiatu (ozn)*, *nazwę powiatu (powiat)*, *siedzibę powiatu (siedziba)*, *typ powiatu (typ)*, *gdzie z oznacza ziemski, g–grodzki i s–stołeczny*.

**Przykład:**

ozn	powiat	siedziba	typ
DBA	walbrzyski	Walbrzych	z
DBL	boleslawiecki	Boleslawiec	z

Plik o nazwie `uslugi.txt` zawiera w każdym wierszu: *NIP firmy (NIP)*; *numer rejestracyjny pojazdu rozbity na oznaczenie literowe (ozn) i część alfanumeryczną (nr)*; *rodzaj świadczonej usługi (rodzaj\_uslugi)*, *litera L oznacza leasing, litera W oznacza wynajem; miesięczną ratę opłaty za usługę (RATA)*.

**Przykład:**

NIP	ozn	nr	rodzaj_uslugi	rata
6727559092	WW	323W	L	1240
3866883575	LU	16794	L	3022
1165493421	HP	LL545	L	1770

Plik o nazwie `nip_firm.txt` zawiera: w każdym wierszu: *NIP firmy (NIP)* i *nazwę firmy (FIRMA)*.

**Przykład:**

NIP	firma
2394072755	ELINK
1599095267	ZONAN
3357631311	GAZPOL

Korzystając z danych zawartych w tych plikach oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź umieść w pliku `wyniki6.txt`, poprzedzając ją oznaczeniem odpowiedniego podpunktu od a) do e).

- Podaj sumaryczną miesięczną kwotę, która wpływa do GoldCar z tytułu opłat za wynajem oraz sumaryczną miesięczną kwotę za usługę leasingowania pojazdów.
- Wykonaj zestawienie kompletnych numerów rejestracyjnych (ozn oraz nr) pojazdów wynajętych lub wziętych w leasing przez firmę „BARTEX”. Zestawienie posortuj nierosnąco według pola nr.
- Podaj nazwę firmy, która wzięła w leasing najwięcej pojazdów od GoldCar oraz liczbę tych pojazdów.  
Uwaga: Jest tylko jedna taka firma.
- Podaj nazwy powiatów ziemskich, w których klienci firmy GoldCar rejestrowali swoje pojazdy.
- Dla firm, które zarejestrowały swoje pojazdy w powiecie o nazwie **Konin**, wykonaj zestawienie, zawierające nazwy firm korzystających z usług GoldCar oraz średnią miesięczną ratę dla każdej firmy. Średnią miesięczną ratę zaokrąglaj do jednego grosza.

Do oceny oddajesz plik(i) o nazwie(ach) .....,  
tu wpisz nazwę(y) pliku(ów)  
zawierający(e) komputerową(e) realizację(e) Twoich obliczeń oraz plik tekstowy o nazwie `wyniki6.txt` z odpowiedziami do podpunktów a, b, c, d, e.

### Sprawdzane umiejętności

W zadaniu były badane umiejętności z II i III obszaru standardów. Zdający:

- analizuje problem i zbiór danych, którego rozwiązanie wymaga zaprojektowanie i utworzenie relacyjnej bazy danych (tabeli i relacji między nimi) z uwzględnieniem zawartych informacji (III.3),
- wyszukuje informacje w bazie danych, stosując różne techniki (w tym zapytania) oraz zastosowanie metod optymalizujących wyszukiwanie (indeksowanie) (II.1).

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
5.a.	filtruje w jednej tabeli.	2	0,79	0,84	–	0,72
5.b.	wyszukuje informacje z dwóch tabel, grupuje i sortuje.	2	0,70	0,74	–	0,62
5.c.	wyszukuje informacje z dwóch tabel, grupuje i sortuje.	2	0,57	0,64	–	0,46
5.d.	grupuje, filtruje.	2	0,42	0,48	–	0,33
5.e.	filtruje, stosuje funkcję średnią, podaje wynik z odpowiednią dokładnością.	2	0,51	0,60	–	0,36

**Przykładowy poprawny zapis rozwiązania:**

a) odp.:

wynajem: 400 713

leasing: 391 790

```
SELECT Sum(Uslugi.rata) AS [Summaryczna kwota], Uslugi.rodzaj_uslugi FROM
Uslugi GROUP BY Uslugi.rodzaj_uslugi;
```

b) odp.:

**ID NR**

PN 3661T	PN 16834
PN 3651R	PN 16832
PN 3641B	PN 16830
PN 3631H	PN 16828
PN 3621I	PN 16826
PN 16838	PN 16824
PN 16836	PN 16822

```
SELECT Uslugi.ozn, Uslugi.nr FROM Tablice INNER JOIN (Nip_firm INNER JOIN
Uslugi ON Nip_firm.NIP = Uslugi.NIP) ON Tablice.ozn = Uslugi.ozn WHERE
((Nip_firm.firma)="BARTEX")) GROUP BY Uslugi.ozn, Uslugi.nr ORDER BY
Uslugi.nr DESC;
```

c)

odp.: WROCBUD2, liczba pojazdów 13

```
SELECT Nip_firm.firma, Uslugi.rodzaj_uslugi, Count(Nip_firm.firma) AS
PoliczOffirma FROM Nip_firm INNER JOIN Uslugi ON Nip_firm.NIP = Uslugi.NIP
GROUP BY Nip_firm.firma, Uslugi.rodzaj_uslugi HAVING
(((Uslugi.[rodzaj_uslugi])="L")) ORDER BY Count(Nip_firm.firma) DESC;
```

d) odp.:

bedzinski	kluczborski	pilski
bialostocki	koscianski	siemiatycki
braniewski	leczynski	tarnobrzescski
buski	lubelski	tarnogorski
elblaski	nyski	tarnowski
elcki	olawski	turecki
gorowski	olsztynski	zamojski
jeleniogorski	opolski	

```
SELECT DISTINCT Tablice.powiat, Tablice.rodzaj FROM Tablice INNER JOIN
(Nip_firm INNER JOIN Uslugi ON Nip_firm.NIP=Uslugi.NIP) ON
Tablice.ozn=Uslugi.ozn GROUP BY Tablice.powiat, Tablice.rodzaj HAVING
(((Tablice.rodzaj) Like "z"));
```

e) odp.:

**FIRMA      Średnia miesięczna rata**

BARTEX    2 325,64 zł

WIET-IND 2 556,43 zł

REFAMUR 2 590,13 zł

```
SELECT Tablice.siedziba, Nip_firm.firma, Avg(Uslugi.rata) AS ŚredniaOfrata
FROM Tablice INNER JOIN (Nip_firm INNER JOIN Uslugi ON
ip_firm.NIP=Uslugi.NIP) ON Tablice.ozn=Uslugi.ozn GROUP BY
Tablice.siedziba, Nip_firm.firma HAVING (((Tablice.siedziba) Like
"konin"));
```

**Komentarz:**

Zadanie miało charakter bazodanowy. Dane do zadania umieszczono w trzech plikach opisujących samochody wynajmowane od firmy leasingowej. Pliki te odpowiadają tabelom relacyjnej bazy danych (tabelom powiązanych poprzez pola *NIP* firmy oraz literowe oznaczenie powiatu *ozn*). Skonstruowanie kwerend w dwóch pierwszych podpunktach okazało się dla zdających stosunkowo proste. Kwerenda w punkcie a) odnosiła się tylko do jednej tabeli, wystarczyło odfiltrować rekordy z samochodów leasingowanych i podsumować raty za tę usługę, a następnie to samo zrobić z samochodami wynajmowanymi. W punkcie b) należało wyłuskać informacje z dwóch tabel i odfiltrować numery rejestracyjne firmy BARTEX. Punkty c) d) i e) okazały się trudniejsze dla zdających. W c) należało policzyć liczbę pojazdów branych w leasing przez firmy i wyszukać maksimum. Tu pojawiały się błędy polegające na pomijaniu przez zdających informacji, że chodzi o pojazdy leasingowane, a nie o wszystkie. W punkcie d) zdający nieprawidłowo podawali wszystkie powiaty (zamiast tylko ziemskie) lub wyszukiwali informacje tylko w jednej tabeli *tablice*, bez połączenia jej relacją z tabelą *usługi*. W punkcie e) należało połączyć wszystkie trzy tabele i skorzystać z funkcji agregujących (*średnia*). Podpunkt ten charakteryzował się wysoką frakcją opuszczeń. Zdarzało się, choć dużo rzadziej niż w latach ubiegłych, że uczniowie zamieszczali tylko plik tekstowy z odpowiedziami, nie dołączając plików realizacji obliczeń.

## Wnioski wynikające z analizy jakościowej zadań

Arkusze egzaminacyjne z informatyki składały się z różnych rodzajów zadań. Zawierały zadania otwarte oraz zamknięte, zadania wymagające umiejętności programistycznych, konstruowania i analizowania algorytmów, posługiwania się programami narzędziowymi. Różnice między arkuszami z poziomu podstawowego i rozszerzonego adekwatnie odzwierciedlały odpowiednie różnice w standardach wymagań egzaminacyjnych. Zadania sformułowane były precyzyjnie i krótko. Można je było rozwiązać różnymi metodami, przy pomocy dostępnych narzędzi, decyzja wyboru metody i programu należała do zdającego. Poruszane problemy nie wykraczały poza standardy wymagań egzaminacyjnych, wynikające z podstawy programowej. Wyniki uzyskane przez maturzystów były różnorodne, co wskazuje na duże zróżnicowanie poziomu ich wiedzy. Dostatecznie istotne różnice w wynikach występowały pomiędzy typami szkół (liceum ogólnokształcącym i technikum), a także pomiędzy konkretnymi szkołami, które kończyli maturzyści. W niektórych szkołach zdający doskonale radzili sobie z zadaniami, w innych wyniki były znacząco niższe. Cieszą dane statystyczne dotyczące wyników egzaminu maturalnego z informatyki. Średnia zdających na poziomie rozszerzonym w kolejnych latach rosła (w 2010 – 47,2%, w 2011 – 56%, w 2012 – 56%), co świadczy o dokonywaniu przez maturzystów dojrzałych decyzji odnośnie wybieranych przedmiotów dodatkowych i prawidłowym przygotowywaniu się do egzaminu. Wzrasta również zainteresowanie egzaminem z informatyki na maturze (w 2010 – 2742 zdających, w 2011 – 3070 zdających, w 2012 – 3799 zdających).

Poniżej przedstawiono kilka wskazówek, jak poprawić wyniki osiągane przez maturzystów i w jaki sposób ukierunkować przygotowania uczniów do przyszłorocznej matury.

W dużej mierze wyniki osiągane przez uczniów na egzaminie zależą od zastosowania przez nauczyciela informatyki prawidłowej ścieżki przygotowań do tego egzaminu. W momencie podejmowania decyzji o wyborze przedmiotu dodatkowego ważne jest, aby nauczyciel zapoznał swoich uczniów z dokumentami i materiałami, które mogą im pomóc w przygotowaniu do egzaminu maturalnego. Są to:

- informator maturalny (zawiera opis struktury i formy egzaminu, wymagania egzaminacyjne oraz procedury przeprowadzania egzaminu maturalnego z informatyki),
- komentarz do zadań egzaminacyjnych z informatyki wydawany przez OKE lub CKE (w publikacji z przedmiotami matematyczno–przyrodniczymi). Komentarz ten zawiera przykładowe rozwiązania zadań oraz opis najczęstszych błędów popełnianych przez zdających, które należy eliminować w procesie nauczania,
- arkusze egzaminacyjne z lat poprzednich, udostępnione na stronie CKE.

Warto, aby nauczyciel w czasie zajęć z informatyki uwzględniał wymagania egzaminacyjne zawarte w informatorze, stosował w praktyce szkolnej zasady oceniania zadań stosowane na egzaminie, zwracał uwagę i eliminował błędne nawyki uczniów oraz niestaranność w rozwiązywaniu zadań (nieuważne czytanie poleceń, pomijanie pewnych elementów polecenia, podawanie wyników z inną dokładnością niż żądana w zadaniu, brak opisów na wykresach itp.). Ważne jest, aby uczeń już na sprawdzianach przyzwyczajał się do formy prezentowania rozwiązania (dołączał zarówno plik tekstowy z odpowiedziami jak i pliki komputerowej realizacji rozwiązania). Do tej pory (choć zdecydowanie rzadziej niż w latach ubiegłych) zdarzają się prace bez dołączonej komputerowej realizacji obliczeń, co powoduje, że uczeń otrzymuje za zadanie 0 punktów. Co charakterystyczne, zdający nadal mają najwięcej problemów z zadaniami dotyczącymi algorytmiki i programowania, a przecież są to dla informatyka umiejętności kluczowe i dlatego tym zagadnieniom należy poświęcić najwięcej czasu podczas przygotowań. Poprawiły się natomiast umiejętności uczniów



w zakresie wykorzystania dostępnych narzędzi informatycznych, tj. arkusza kalkulacyjnego czy aplikacji do obsługi baz danych.

Co roku w lipcu (10 miesięcy przed egzaminem) na stronie Centralnej Komisji Egzaminacyjnej ukazuje się komunikat Dyrektora CKE o egzaminie maturalnym z informatyki, w którym podane są dopuszczone środowiska i języki programowania na najbliższą sesję egzaminacyjną. Dobrze byłoby, gdyby uczeń od początku pracował z programami, których będzie używał podczas egzaminu. Ich biegła znajomość może okazać się niezwykle przydatna podczas egzaminu, kiedy uczeń ma limitowany czas na rozwiązanie zadań i szkoda go poświęcać na zapoznawanie się ze środowiskiem.