

Informatyka

Egzamin maturalny z informatyki odbył się w całym kraju w dniu 18 maja 2010 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-102) oraz *Arkusz II* (MIN-P2_1P-102)
- rozszerzonym – *Arkusz I* (MIN-R1_1P-102) oraz *Arkusz II* (MIN-R2_1P-102)

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, a *Arkusz II* 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, a *Arkusz II* 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

Część I

Zadanie 1. Szyfrowanie (8 pkt)

Poniższy algorytm szyfruje słowo s przy pomocy pewnego szyfru przestawieniowego. Zasyfrowane słowo zostaje zapisane w zmiennej w .

Algorytm

1. $i \leftarrow 1$, $w \leftarrow "$, $k \leftarrow 2$
2. dopóki $i \leq k$ wykonuj
 - $j \leftarrow i$
 - dopóki $j \leq \text{długość}(s)$ wykonuj:
 - $w \leftarrow w \bullet s[j]$
 - $j \leftarrow j + k$
 - $i \leftarrow i + 1$

Objaśnienia: " oznacza słowo puste, $s[j]$ oznacza j -ty znak w słowie s (**numeracja rozpoczyna się od 1**), \bullet oznacza sklejanie dwóch słów (ewentualnie sklejanie słowa i znaku), \leftarrow oznacza instrukcję przypisania.

- a) Zapisz wynik działania powyższego algorytmu dla słów ARKA i MOTOR.

.....

1c.

1. $i \leftarrow 1, w \leftarrow "$
2. dopóki $i < \text{długość}(s)$ wykonuj:
 - $w \leftarrow w \bullet s[i+1]$
 - $w \leftarrow w \bullet s[i]$
 - $i \leftarrow i+2$
3. jeśli $i = \text{długość}(s)$, to $w \leftarrow w \bullet s[i]$

Komentarz:

Wskaźnik łatwości zadania 0,38 klasyfikuje zadanie jako trudne. Najwięcej problemów zadającym sprawił podpunkt c), w którym należało zapisać w postaci algorytmu przedstawioną w treści zadania metodę szyfrowania. Zdający, wzorując się na algorytmie podanym w pierwszej części zadania, konstruowali algorytm, w którym jednak pojawiały się błędy: nieprawidłowa inkrementacja liczników, zapętlenie się algorytmu, nieprawidłowe działanie dla haseł o parzystej lub nieparzystej liczbie znaków. Zdarzały się również rozwiązania prezentujące metodę szyfrowania, której nie można jednak zaliczyć do algorytmów, a raczej do przepisu słownego np. „zlicz ilość znaków w słowie”, „dziel słowo przez dwa”, „zamień kolejność znaków” itp.

Zadanie 2. Rozkład liczby (7 pkt)

Rozkładem na czynniki pierwsze liczby całkowitej większej od 1 nazywamy przedstawienie tej liczby w postaci iloczynu czynników pierwszych (liczb pierwszych). Jeżeli dana liczba jest liczbą pierwszą, to w jej rozkładzie występuje tylko ona sama.

Przykłady:

$$24 = 2 \cdot 2 \cdot 2 \cdot 3$$

$$20 = 2 \cdot 2 \cdot 5$$

$$19 = 19$$

- a) Podaj rozkład na czynniki pierwsze następujących liczb całkowitych:

Liczba	Rozkład na czynniki pierwsze
63	
184	
277	

- b) Ułóż algorytm (w postaci listy kroków, schematu blokowego lub w wybranym języku programowania), który dla liczby całkowitej n ($n > 1$) podaje wszystkie jej czynniki pierwsze występujące w rozkładzie.

Specyfikacja

Dane: liczba całkowita n ($n > 1$)

Wynik: wszystkie czynniki pierwsze liczby n

Przykłady:

Dla $n = 24$ poprawnym wynikiem jest 2, 2, 2, 3.

Dla $n = 19$ poprawnym wynikiem jest 19.

[illegible]

Trudny dla zdających w tym zadaniu okazał się podpunkt b), należało zapisać algorytm, który dla liczby całkowitej n ($n > 1$) podaje wszystkie jej czynniki pierwsze występujące w rozkładzie. Najczęstsze błędy to brak inicjacji zmiennych wartościami początkowymi, wypisywanie wszystkich dzielników liczby n zamiast rozkładu na czynniki pierwsze, dzielenie podanej liczby n przez kilka początkowych liczb pierwszych np. podzielić n przez 2, jeżeli wynik jest liczbą całkowitą, wypisz 2, podzielić n przez 3 itd. Zdarzały się rzadkie rozwiązania z użyciem rekurencji.

Zadanie 3. Test (5 pkt)

Dla następujących zdań **zaznacz znakiem X** właściwe odpowiedzi.

Uwaga: W każdym podpunkcie poprawna jest tylko jedna odpowiedź.

- a) Wiadomo, że kodem ASCII małej litery „a” jest 97. Kodem małej litery „d” jest liczba
- ☐ (01100100)₂
 - ☐ (01011010)₂
 - ☐ (01101110)₂
- b) Zbiór programów, które zarządzają pracą komputera, to
- ☐ oprogramowanie edukacyjne.
 - ☐ system operacyjny.
 - ☐ pakiet programów biurowych.
- c) Popularnym językiem programowania jest
- ☐ Kawa.
 - ☐ Java.
 - ☐ Agawa.
- d) Który parametr **nie dotyczy** charakterystyki dysku twardego?
- ☐ Pojemność liczona np. w GB.
 - ☐ Czas zapisu i odczytu 1 kB danych liczony w cyklach rozkazowych procesora.
 - ☐ Liczba obrotów talerzy dysku na minutę (obr/min).
- e) Dany jest następujący algorytm:
1. $n \leftarrow 6, i \leftarrow 1, s \leftarrow 0$
 2. dopóki $i \leq n$ wykonuj
 - $s \leftarrow s + i$
 - $i \leftarrow i + 1$
 3. $s \leftarrow s + 1$

Uwaga: \leftarrow oznacza instrukcję przypisania.

Po wykonaniu tego algorytmu wartości zmiennych s, i są odpowiednio równe

- ☐ 21, 6.
- ☐ 22, 7.
- ☐ 21, 7.

Część II**Sprawdzane umiejętności**

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

- zna sposoby reprezentowania informacji w komputerze,
- zna podstawowe funkcje systemu komputerowego,
- zna typowe narzędzia informatyczne,
- zna funkcjonowanie komputera i jego części składowych,
- zna pojęcie algorytmu i różne sposoby jego zapisu, wyodrębnia składowe algorytmu.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a.	podaje kod ASCII dla litery <i>d</i> i zapisuje liczbę 100 w systemie binarnym	1	0,78	0,75	-	0,79
3.b.	wskazuje oprogramowanie, które zarządza pracą komputera	1	0,98	0,98	-	0,98
3.c.	zna języki programowania	1	0,99	0,98	-	1,00
3.d.	zna parametry dysku twardego	1	0,87	0,77	-	0,90
3.e.	analizuje algorytm, podaje wartości zmiennych po wykonaniu algorytmu	1	0,56	0,54	-	0,58

Poprawny zapis rozwiązania:
a1, b2, c2, d2, e2

Komentarz:
Zadanie miało charakter testu wyboru, sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Podpunkty a) b) c) i d) okazały się dla zdających łatwe i bardzo łatwe, natomiast e) – umiarkowanie trudny. W podpunkcie a) należało policzyć kod ASCII dla litery *d*, wiedząc że dla *a* kod wynosi 97, a następnie przekonwertować liczbę 100 z zapisu dziesiętnego na binarny. W podpunkcie e) zdający powinni prześledzić wartości zmiennych *s* i *i*. Przy ostatnim przebiegu pętli: dopóki $i \leq n$ wykonuj zmienna $s=21$, zaś $i=7$. W kroku 3. *s* zwiększa się o 1, więc po wykonaniu algorytmu $s=22$. W rozwiązaniu często pojawiała się nieprawidłowa odpowiedź trzecia. Zdający prawdopodobnie pomijali krok 3. algorytmu.

Zadanie 4. Palindromy (10 pkt)

Palindromem nazywamy słowo, które czytane od lewej i od prawej strony jest takie samo.

Na przykład palindromami są słowa:

JABFDFBAJ

HAJAHAJAH

ABBA

Słowo JANA nie jest palindromem.

W pliku dane.txt umieszczono w kolejnych wierszach 1000 słów o długościach od 2 do 25 znaków, składających się z wielkich liter A, B, C, D, E, F, G, H, I, J. **Napisz program**, który przegląda słowa zapisane w pliku dane.txt i wypisuje te z nich, które są **palindromami**, po jednym w wierszu. Kolejność wypisywania palindromów powinna być taka sama jak w pliku z danymi. Wyniki zapisz w pliku zadanie4.txt.

Do oceny oddajesz plik zadanie4.txt oraz plik(i)

tu wpisz nazwę(y) pliku / plików

zawierający(e) tekst źródłowy programu.

Sprawdzane umiejętności

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

- posługuje się kompilatorem wybranego języka programowania,
- formułuje informatyczne rozwiązanie problemu przez dobór algorytmu oraz odpowiednich

struktur danych i realizuje je w wybranym języku programowania						
Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
4.	pisze program, sprawdzający czy podany łańcuch znaków jest palindromem	10	0,16	0,13	-	0,17
<p>Przykładowy poprawny zapis rozwiązania:</p> <p>odp.:</p> <p>CBJBC</p> <p>AABBHHIHHBBA</p> <p>CCJGJCC</p> <p>FDDFFFD</p> <p>BCACHCACB</p> <p>DFDCCDFD</p> <p>DFFFD</p> <p>AABBAA</p> <p>AABAABAABA</p> <p>AJAJAJAJA</p> <p>FAFBDCEAAECDBFAF</p> <p>CEDDIGIDDEC</p> <p>ABFACDDABADDCAFBA</p> <p>EBE</p> <p>EABAE</p> <p>FDFBCFFCBFDF</p> <p>FDDEDDF</p> <p>FAAAF</p> <p>AABBAA</p> <p>EECEE</p> <p>DDFFEEJGJEEFFDD</p> <p>BCEABDECCEDBAECB</p> <p>FEDFBBGGBBFDEF</p> <p>BDDDBCEADDBFCCFBDDAECBDD</p> <p>BAAJDJAAB</p> <p>BBFFGJGFFBB</p> <p>CBJGJBC</p> <p>BDDDB</p> <p>AEFDCEFEFECDFEA</p> <p>ABBCCBBA</p> <p>ABCDEFHJIHGFEDCBA</p> <p>BADADAB</p> <p>FGHAHGF</p> <p>AAABEDEBAAA</p> <p>CFCFC</p> <p>HJHHJHHJH</p> <p>DCFCD</p> <p>BDCAEEIIEEACDB</p> <p>BDDFBFAFABFDD</p> <p>ADAADA</p> <p>ABCGCBA</p> <p>GABAG</p> <p>przykładowe rozwiązanie w C++:</p> <pre>#include<iostream> #include<string> #include<fstream> using namespace std; string odwroc(string s) {</pre>						

```
string wynik;
for(int i=0; i<s.size(); i++)
    wynik = s[i] + wynik;
return wynik;
}

main()
{
    string s;
    ifstream plik("dane.txt");
    ofstream wynik("zadanie4.txt");

    while(!plik.eof())
    {
        plik>>s;
        if(s==odwroc(s))
            wynik<<s<<endl;
    }

    plik.close();
    wynik.close();
    cout<<"Aby zakonczyc prosze nacisnac dowolny klawisz...";
    getchar();
    return 0;
}
```

przykładowe rozwiązanie w języku Java:

```
import java.io.*;

public class Program4{

    public static boolean czyPalindrom(String s){

        for(int i=0;i<=(int) (s.length()/2);i++)
            if(s.charAt(i)!=s.charAt(s.length()-i-1))
                return false;
        return true;
    }

    public static void main(String[] args){

        RandomAccessFile raf;

        try{
            raf=new RandomAccessFile("dane.txt","r");
        }
        catch(FileNotFoundException e){
            System.out.println(e);
            return;
        }

        BufferedWriter bw;

        try{
            bw=new BufferedWriter(new FileWriter("zadanie4.txt"));
        }
        catch(IOException e){
            System.out.println(e);
            return;
        }
    }
}
```



```
String line;

try{

    while((line=raf.readLine())!=null){
        if(czyPalindrom(line)){
            bw.write(line,0,line.length());
            bw.newLine();
        }
    }

    raf.close();
    bw.close();

}
catch(IOException e){
    System.out.println(e);
    return;
}

}
```

Komentarz:

Zadanie 4 miało charakter programistyczny. Wyróżniało się tym, iż w treści wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego. Wśród prawidłowych rozwiązań pojawiały się dwie metody: pierwsza polegała na odwróceniu łańcucha znaków i porównaniu z łańcuchem źródłowym; druga na porównywaniu znaków pierwszego i ostatniego, drugiego i przedostatniego itd. do połowy długości łańcucha znaków. Sprawdzenie, czy podany łańcuch znaków jest palindromem, należy do typowych algorytmów w tekstach. Wysoka frakcja opuszczeń tego zadania (19,4%) pozwala przypuszczać, że zdający nie znają prostych klasycznych algorytmów, mają problemy z zapisywaniem algorytmów w postaci prawidłowo działających programów komputerowych albo z założenia pomijają zadanie programistyczne. Należy tu zwrócić uwagę, że tworzenie programów komputerowych to kluczowe umiejętności informatyczne.

Zadanie 5. Upusty (10 pkt)

Producenci A i B sprzedają pewien towar po 12,00 zł za sztukę. Producent A daje odbiorcom 15% upustu przy zakupie do 500 sztuk oraz 25% upustu przy zakupie powyżej 500 sztuk. Natomiast producent B dla odbiorców przygotował 10% upustu przy zakupie do 300 sztuk, 10% upustu przy zakupie do 300 sztuk oraz 25% upustu od każdej sztuki powyżej 300 – do 600 sztuk, natomiast przy zakupie powyżej 600 sztuk oferuje upust 35% od każdej zakupionej sztuki.

Dokładne reguły wyznaczania kwoty do zapłacenia, w zależności od liczby sztuk towaru, są następujące:

x – liczba sztuk, k – koszt zakupu.

Producent A:

$$x \leq 500 \quad k = x * (1 - 0,15) * 12$$

$$x > 500 \quad k = x * (1 - 0,25) * 12$$

Producent B:

$$x \leq 300 \quad k = x * (1 - 0,1) * 12$$

$$\begin{aligned} 300 < x \leq 600 & \quad k = 300 * (1 - 0,1) * 12 + (x - 300) * (1 - 0,25) * 12 \\ x > 600 & \quad k = x * (1 - 0,35) * 12 \end{aligned}$$

Towar pakowany jest po 10 sztuk, dlatego wartości x w powyższym opisie i w odpowiedziach na poniższe pytania mogą być tylko wielokrotnościami liczby 10. W obliczeniach przyjmij, że x jest liczbą z przedziału $[10, 1000]$.

Wykorzystując dostępne narzędzia informatyczne, wykonaj potrzebne obliczenia i odpowiedz na poniższe pytania. Odpowiedzi umieść w pliku tekstowym `zadanie5.txt`, każdą odpowiedź poprzedź literą oznaczającą stosowny podpunkt.

- W jakich przedziałach wartości koszt zakupu x sztuk towaru jest niższy u producenta A?
- Przy jakiej liczbie sztuk towaru nie ma znaczenia, u którego producenta hurtownia robi zakupy?
- Jaka jest największa liczba sztuk, dla której koszt zakupu u producenta A jest mniejszy niż koszt zakupu 500 sztuk u producenta B?
- Jaka jest największa liczba sztuk, dla której koszt zakupu u producenta B jest mniejszy niż koszt zakupu 600 sztuk u producenta A?

Do oceny oddajesz plik(i) o nazwie(ach), zawierający(e)
tu wpisz nazwę(y) pliku / plików
komputerową(e) realizację(e) Twoich obliczeń oraz plik tekstowy `zadanie5.txt`,
zawierający odpowiedzi do podpunktów a), b), c) i d) zadania.

Sprawdzane umiejętności

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

- dobiera właściwy program (użytkowy lub własnoręcznie napisany) do rozwiązywanego zadania,
- dobiera metody i narzędzia informatyczne do wykonywanych zadań.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
5.a.	oblicza koszty zakupu towaru u producentów A i B	4	0,35	0,38	-	0,33
5.b.	sprawdza, dla jakiej liczby sztuk koszty zakupu towaru są równe u obu producentów	2	0,56	0,67	-	0,52
5.c.	przeprowadza symulację	2	0,34	0,39	-	0,31
5.d.	przeprowadza symulację	2	0,36	0,41	-	0,34

Przykładowy poprawny zapis rozwiązania:

a) odp.: $\langle 10,440 \rangle$ i $\langle 510,600 \rangle$

Microsoft Excel - Zadanie5obliczenia.xls [Tylko do odczytu]

Plik Edycja Widok Wstaw Format Narzędzia Dane Okno Pomoc

Arial 10 B

B2 $=JEŻELI(A2 \leq 500; A2 * (1 - 0,15) * 12; A2 * (1 - 0,25) * 12)$

	A	B	C	D	E	F
1	x	koszt u producenta A	koszt u producenta B	A<B	A=B	
2	10	102	108	PRAWDA		
3	20	204	216	PRAWDA		
4	30	306	324	PRAWDA		
5	40	408	432	PRAWDA		
6	50	510	540	PRAWDA		
7	60	612	648	PRAWDA		
8	70	714	756	PRAWDA		

Arkusz1 / Arkusz2 / Arkusz3 /

Rysuj Autokształty

Gotowy NUM

$=JEŻELI(A2 \leq 500; A2 * (1 - 0,15) * 12; A2 * (1 - 0,25) * 12)$

$=JEŻELI(A2 \leq 300; A2 * (1 - 0,1) * 12; JEŻELI(A2 > 600; A2 * (1 - 0,35) * 12; 300 * (1 - 0,1) * 12 + (A2 - 300) * (1 - 0,25) * 12))$

$=JEŻELI(B2 < C2; "PRAWDA"; "")$

b) odp.: 450 sztuk

$=JEŻELI(B2 = C2; "PRAWDA"; "")$

c) odp.: 550 sztuk

Microsoft Excel - Zadanie5.xls

Plik Edycja Widok Wstaw Format Narzędzia Dane Okno Pomoc

Arial 11 B

A56 550

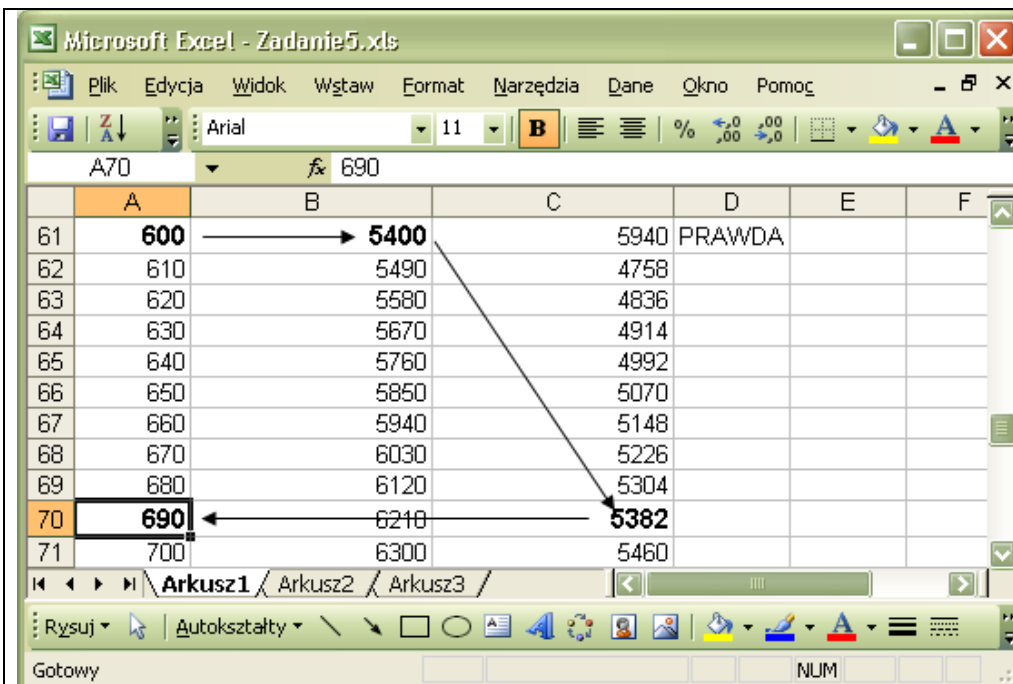
	A	B	C	D	E	F
50	490	4998	4950			
51	500	5100	5040			
52	510	4590	5130	PRAWDA		
53	520	4680	5220	PRAWDA		
54	530	4770	5310	PRAWDA		
55	540	4860	5400	PRAWDA		
56	550	4950	5490	PRAWDA		
57	560	5040	5580	PRAWDA		

Arkusz1 / Arkusz2 / Arkusz3 /

Rysuj Autokształty

Gotowy NUM

d) odp.: 690 sztuk



	A	B	C	D	E	F
61	600	5400	5940	PRAWDA		
62	610	5490	4758			
63	620	5580	4836			
64	630	5670	4914			
65	640	5760	4992			
66	650	5850	5070			
67	660	5940	5148			
68	670	6030	5226			
69	680	6120	5304			
70	690	6210	5382			
71	700	6300	5460			

Komentarz:

Zadanie 5 okazało się dla zdających trudne. Maturzyści rozwiązywali je w arkuszu kalkulacyjnym, rzadziej decydowali się na napisanie programu. Najczęstszym błędem było wykonywanie obliczeń dla liczby towarów podawanej z dokładnością do 1, zamiast do 10. Uczniowie nie brali pod uwagę zapisu w treści zadania: „Towar pakowany jest po 10 sztuk, dlatego wartości x w powyższym opisie i w odpowiedziach na poniższe pytania mogą być tylko wielokrotnościami liczby 10.” W podpunkcie a) często podawany był jeden przedział zamiast dwóch. W wielu rozwiązaniach pojawiały się błędy obliczeniowe, szczególnie w formule obliczającej ceny towarów u producenta B.

Zadanie 6. Noworodki (10 pkt)

Pliki `noworodki.txt` oraz `mamy.txt` zawierają dane o dzieciach i ich matkach.

W pliku `noworodki.txt` każdy wiersz zawiera następujące informacje o jednym dziecku, rozdzielone znakami odstępu: *identyfikator*, *pleć* (c – córka, s – syn), *imię*, *data urodzenia*, *waga [g]*, *wzrost [cm]* oraz *identyfikator matki*.

Przykład:

1 c Agnieszka 20-lis-1999 2450 48 33

W pliku `mamy.txt` każdy wiersz zawiera informacje o jednej kobiecie, rozdzielone znakami odstępu: *identyfikator matki*, *imię*, *wiek*.

Przykład:

1 Agata 25

Identyfikator matki z pliku `noworodki.txt` odpowiada identyfikatorowi w pliku `mamy.txt`.

Wykorzystując dane zawarte w plikach `mamy.txt` i `noworodki.txt` oraz dostępne narzędzia informatyczne, wykonaj poniższe polecenia. Odpowiedzi umieść w pliku tekstowym `zadanie6.txt`, każdą odpowiedź poprzedź literą oznaczającą stosowny podpunkt.

- a) Podaj imię i wzrost najwyższego chłopca oraz imię i wzrost najwyższej dziewczynki.
Uwaga: Jest tylko jeden taki chłopiec i tylko jedna taka dziewczynka.
- b) W którym dniu urodziło się najwięcej dzieci? Podaj datę i liczbę dzieci.
Uwaga: Jest tylko jeden taki dzień.
- c) Podaj imiona kobiet w wieku poniżej 25 lat, które urodziły dzieci o wadze powyżej 4000 g.
- d) Podaj imiona i daty urodzenia dziewczynek, które odziedziczyły imię po matce.
- e) W pliku `noworodki.txt` zapisane są informacje o narodzinach bliźniąt. Bliźnięta można rozpoznać po tej samej dacie urodzenia i tym samym identyfikatorze matki. Pamiętaj, że przykładowo Jacek i Agatka oraz Agatka i Jacek to ta sama para. Możesz założyć, że w danych nie ma żadnych trojaczków, czworaczków, itd. Podaj daty, w których urodziły się bliźnięta.

Do oceny oddajesz plik(i) o nazwie(ach), zawierający(e)
 tu wpisz nazwę(y) pliku / plików
 komputerową(e) realizację(e) Twoich obliczeń oraz plik tekstowy `zadanie6.txt`
 zawierający odpowiedzi do podpunktów a), b), c), d) i e) zadania.

Sprawdzane umiejętności W zadaniu były badane umiejętności z III obszaru standardów. Zdający: <ul style="list-style-type: none"> projektuje relacyjne bazy danych i wykorzystuje do ich realizacji system bazy danych, stosuje metody wyszukiwania i przetwarzania informacji w relacyjnych bazach danych. 						
Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
6.a.	projektuje relacyjną bazę danych, wyszukuje informacje	2	0,82	0,88	-	0,79
6.b.	projektuje relacyjną bazę danych, wyszukuje informacje	2	0,69	0,79	-	0,64
6.c.	projektuje relacyjną bazę danych, wyszukuje informacje	2	0,56	0,70	-	0,51
6.d.	projektuje relacyjną bazę danych, wyszukuje informacje, porządkuje informacje, zapisuje w zadanym formacie	2	0,53	0,63	-	0,50
6.e.	projektuje relacyjną bazę danych, wyszukuje informacje, porządkuje informacje,	2	0,36	0,42	-	0,34
Przykładowy poprawny zapis rozwiązania: a) odp.: dla chłopca – Oskar 62, dla dziewczynki – Maja 61 <pre>SELECT TOP 1 noworodki.imie, noworodki.wzrost, noworodki.plec FROM noworodki WHERE (((noworodki.plec)="s")) ORDER BY noworodki.wzrost DESC; SELECT TOP 1 noworodki.imie, noworodki.wzrost, noworodki.plec FROM noworodki WHERE (((noworodki.plec)="c")) ORDER BY noworodki.wzrost DESC;</pre> b) odp.: urodzenia: 1999-11-20, liczba dzieci: 23 <pre>SELECT TOP 1 noworodki.data_urodzenia, Count(noworodki.identyfikator) AS PoliczOfidentyfikator FROM noworodki GROUP BY noworodki.data_urodzenia</pre>						

```
ORDER BY Count(noworodki.identyfikator) DESC;
```

c) odp.: Barbara, Eliza, Janina, Maria, Marzena, Paulina

```
SELECT mamy.imie, noworodki.waga FROM mamy INNER JOIN noworodki ON
mamy.identyfikator = noworodki.id_matki WHERE (((noworodki.waga)>4000) AND
((mamy.wiek)<25));
```

d) odp.: Anna 1999-11-21 Wiktoria 1999-11-20

```
SELECT noworodki.imie, noworodki.data_urodzenia FROM mamy INNER JOIN
noworodki ON mamy.identyfikator = noworodki.id_matki WHERE
(((noworodki.imie)=[mamy].[imie]));
```

e) odp.:

1999-11-19
1999-11-21
1999-11-22
1999-11-23
1999-12-09
1999-12-12

```
SELECT noworodki.data_urodzenia FROM mamy INNER JOIN noworodki ON
mamy.identyfikator = noworodki.id_matki GROUP BY noworodki.data_urodzenia,
noworodki.id_matki HAVING (((Count(noworodki.identyfikator))>1));
```

Komentarz:

Ostatnie zadanie drugiego arkusza miało charakter bazodanowy, choć można było rozwiązywać je z powodzeniem w arkuszu kalkulacyjnym. Dane do zadania umieszczono w dwóch plikach: `mamy.txt` i `noworodki.txt`. Pliki te odpowiadały dwóm tabelom relacyjnej bazy danych, powiązanych poprzez identyfikator matki. Skonstruowanie kwerend w podpunkcie a) okazało się dla zdających stosunkowo proste. Odnosiły się one tylko do jednej tabeli - `noworodki`, wystarczyło odfiltrować rekordy z chłopcami i wyszukać dziecko o najwyższym wzroście, a następnie wykonać to samo dla dziewczynek. Podpunkty b) c) i d) należały do umiarkowanie trudnych. Kwerenda b) również korzystała jedynie z danych zawartych w tabeli `noworodki` – należało stworzyć zestawienie identyfikator `noworodki` i data urodzenia, a następnie pogrupować dane i policzyć ilość identyfikatorów dla poszczególnych dat oraz wskazać maksimum. W kwerendzie c) zdający powinien do kwerendy wyszukiwanej dopisać kryteria: wiek matki mniejszy niż 25 i waga dziecka większa niż 4000g. W kwerendzie d) należało w polu imię dziecka utworzyć kryterium – równa się imię matki. Najtrudniejszy okazał się podpunkt e). Rozwiązanie polegało na zestawieniu dat urodzenia dzieci i identyfikatorów matki, wypisaniu dni, w których ten sam identyfikator matki pojawiał się dwukrotnie.

POZIOM ROZSZERZONY

Część I

Zadanie 1. Szyfr przestawieniowy (7 pkt)

Szyfrowanie przestawieniowe jest klasyczną metodą szyfrowania, polegającą na zmianie kolejności liter w szyfrowanym tekście. Często używa się reguł zamiany opartych na różnych figurach geometrycznych – w tym zadaniu użyjemy kwadratu. Szyfrowanie będzie polegało na wprowadzeniu tekstu do kwadratowej tablicy szyfrującej o wymiarach $n \times n$ po kolei wierszami, a następnie odczytaniu tekstu z tablicy kolumnami od lewej do prawej. Wymiar n tablicy jest najmniejszą liczbą, przy której tekst zmieści się w całości w kwadracie $n \times n$.

kierunek wprowadzania
szyfrowanego tekstu →

↓
kierunek odczytywania
zaszyfrowanego tekstu

Załóżmy, że tekst **ALGORYTM_PRZESTAWIENIOWY** ma być zaszyfrowany w tablicy kwadratowej. Liczba znaków w tekście do zaszyfrowania jest równa 24, czyli tablica szyfrująca ma wymiary 5×5 . Ostatni element tablicy będzie uzupełniony znakiem odstępu. Tekst zapisujemy do tablicy wierszami.

A	L	G	O	R
Y	T	M	—	P
R	Z	E	S	T
A	W	I	E	N
I	O	W	Y	—

AYRAILTZWOGMEIWO SEYRPTN

- [illegible]

- BTLLTU_ĚL_EOYPM_ĄPJZLCYNDREOKYLI_ZMFO_ĄGJY_Ó_N_DEWFWGISYSII_ŁEI

[illegible]

„BŁĄD JEST PRZYWILEJEM FILOZOFÓW TYLKO GŁUPCY NIE MYLĄ SIĘ NIGDY”

1.c) (rozwiązanie 1)

1. $n := \text{część całkowita}(\sqrt{d})$
2. *dopóki* $n*n < d$ *to* $n \leftarrow n+1$
3. $s \leftarrow n*n$
4. $w \leftarrow 1$
5. *od* $wier=1$ *do* $wier=n$ *wykonuj*
 od $kol=1$ *do* $kol=n$ *wykonuj*
 jeśli $w \leq d$ *to* $tab[wier, kol] \leftarrow tekst[w]$
 w przeciwnym wypadku $tab[wier, kol] := "$ „
 $w \leftarrow w+1$
6. $w \leftarrow 1$
7. *od* $kol=1$ *do* $kol=n$ *wykonuj*
 od $wier=1$ *do* $wier=n$ *wykonuj*
 $tekst[w] \leftarrow tab[wier, kol]$
 $w \leftarrow w+1$

1.c) (rozwiązanie 2)

1. $n := \text{część całkowita}(\sqrt{d})$
2. *dopóki* $n*n < d$ *to* $n := n+1$
3. $s \leftarrow n*n$
4. *od* $i=1$ *do* $i=s$ *wykonuj*
 jeśli $i \leq d$ $tab[i] \leftarrow tekst[i]$
 w przeciwnym wypadku $tab[i] := "$ „
 „
5. *od* $i=1$ *do* $i=n$ *wykonuj*
 $q \leftarrow i$
 od $j=1$ *do* $j=n$ *wykonuj*
 $szyfr[p] = tab[q]$
 $p \leftarrow p+1$
 $q \leftarrow q+n$

Komentarz:

Podpunkty a) i b) były wprowadzeniem do algorytmu. Rozwiązując podpunkt a) uczniowie powinni uświadomić sobie, że zaszyfrowany tekst będzie miał inną długość niż tekst jawny, w szczególnym przypadku – gdy liczba znaków tekstu jawnego jest kwadratem liczby całkowitej – długości obu tekstów będą równe. Zadanie okazało się dla uczniów łatwe, choć zdarzało się, że uczniowie obliczali jedynie \sqrt{d} zapominając, że liczba wierszy/kolumn tablicy powinna być zaokrąglona w górę, do liczby całkowitej nie mniejszej od danej.

W podpunkcie b) uczniowie rozszyfrowywali szyfrogram, aby prześledzić mechanizm szyfru przestawieniowego, przedstawionego w treści zadania.

Rozwiązanie podpunktu c) polegało na zapisaniu algorytmu w postaci listy kroków, schematu blokowego lub w wybranym języku programowania, który szyfruje zadany tekst opisanym sposobem.

Klasycznym (szkolnym) rozwiązaniem jest wpisanie tekstu do tablicy kwadratowej wierszami, uzupełnienie wolnych pól spacjami, a następnie odczytanie tekstu kolumnami (rozwiązanie 1). Można również szyfrować, uzupełniając jawny tekst spacjami do długości $s=n*n$ i odczytując z przesunięciem o n (rozwiązanie 2).

	instrukcja przypisania jest wykonywana w przebiegu algorytmu? Uzasadnia odpowiedź.					
<p>Poprawny zapis rozwiązania / Przykładowy poprawny zapis rozwiązania:</p> <p>2.a) pierwsza luka: $s-l+1$ druga luka: $s+1$</p> <p>2.b) Liczba instrukcji przypisania: $\log_2 1024$ lub $\lceil \log_2 1023 \rceil$ lub 10. Uzasadnienie: w każdym przebiegu pętli zakres tablicy pozostały do sprawdzenia zmniejsza się o połowę, dlatego dla przedziału $a[1 \dots 1023]$ instrukcja przypisania zostanie wykonana 10 razy.</p> <p>Komentarz: Zadanie dotyczyło przeszukiwania binarnego, czyli metody wyszukiwania elementu w posortowanym ciągu, tu w ciągu składającym się początkowo z zer i dalej z jedynek (np. ...000001111...). Sprawdzanie rozpoczyna się od środkowego elementu tablicy $a[s]$, gdzie $s \leftarrow (l+p) \text{div} 2$ i sprawdzane jest, czy element jest już jedynką. Jeżeli tak, to w kolejnym przebiegu algorytmu połowiony będzie przedział od 1 do $s-1$ (odrzuca się prawą połowę tablicy), jeżeli $a[s]=0$ to połowiony będzie przedział od $s+1$ do 1023 (odrzuca się lewą połowę tablicy), a do licznika zer dodaje się liczbę elementów odrzucanego przedziału $s-l+1$. W każdym przebiegu pętli liczba elementów tablicy zmniejsza się o połowę, w związku z tym łatwo obliczyć liczbę instrukcji przypisania wykonywaną w każdym przebiegu tego algorytmu.</p>						

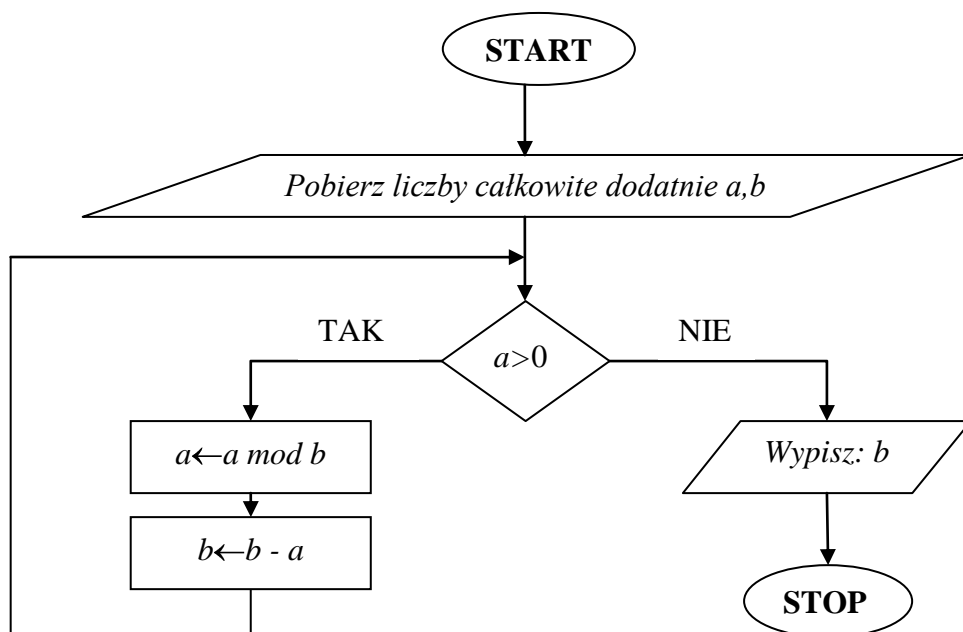
Zadanie 3. (5 pkt)

Podpunkty a) – e) zawierają po trzy stwierdzenia, z których każde jest albo prawdziwe, albo fałszywe. Zdecyduj, które z podanych stwierdzeń są prawdziwe (**P**), a które fałszywe (**F**). **Zaznacz znakiem X** odpowiednią rubrykę w tabeli.

a) Pojedyncza operacja wykonywana na stosie to

	P	F
pobranie pierwszego od dołu elementu.		
usunięcie pierwszego od dołu elementu.		
pobranie pierwszego od góry elementu.		

b) Algorytm



znajduje

	P	F
NWW (a,b) .		
NWD (a,b) .		
liczbę pierwszą większą od a i mniejszą od b .		

c) Liczba 1000_{16} to

	P	F
34522_5		
4096_{10}		
10000_8		

d) Program zapobiegający włamaniom do systemu i kontrolujący pakiety sieciowe to

	P	F
firewall.		
keylogger.		
filtr antyspamowy.		

e) Format plików graficznych dla grafiki rastrowej to

	P	F
BMP.		
JPG.		
GIF.		

Sprawdzane umiejętności

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

- zna wybrane struktury danych i ich realizację
- zna techniki algorytmiczne i algorytmy
- zna systemy liczbowe mające zastosowanie w informatyce
- charakteryzuje typowe narzędzia informatyczne i ich zastosowania
- charakteryzuje sposoby reprezentowania informacji w komputerze

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a	zna i omawia wybrane struktury danych (stos)	1	0,78	0,81	-	0,67
3.b.	zna algorytm Euklidesa zapisany w postaci schematu blokowego	1	0,98	0,77	-	0,89
3.c.	zna i potrafi stosować dowolny pozycyjny system liczbowy oraz zasady konwersji liczb pomiędzy różnymi systemami pozycyjnymi	1	0,99	0,65	-	0,33
3.d	zna zasadę działania programów zabezpieczających system typu firewall, keylogger i filtr antyspamowy	1	0,87	0,88	-	0,89
3.e	rozpoznaje podstawowe formaty plików graficznych	1	0,56	0,57	-	0,44

Poprawny zapis rozwiązania:

- a) FFP
b) FPF
c) FPP
d) PFF
e) PPP

Komentarz:

Zadanie miało charakter testu wyboru, sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Podpunkty a) b) i d) okazały się dla zdających łatwe, natomiast c) i e) – umiarkowanie trudne. Dostatecznie istotne różnice w wynikach występowały pomiędzy typami szkół (liceum ogólnokształcące i technikum), a także pomiędzy konkretnymi szkołami, które kończyli maturzyści. W niektórych szkołach zdający doskonale radzili sobie z tym zadaniem, w innych wyniki były znacząco niższe.

a) Stos jest strukturą liniowo uporządkowanych danych, z których jedynie ostatni element, zwany wierzchołkiem, jest w danym momencie dostępny. W wierzchołku odbywa się dołączanie nowych elementów, również jedynie wierzchołek można usunąć. W związku z tym prawidłowa jest jedynie

odpowiedź ostatnia.

b) W zadaniu należało rozpoznać klasyczny algorytm Euklidesa, służący do znalezienia NWD dwóch liczb a i b , zapisany w postaci graficznej. Jediną prawidłową odpowiedzią była odpowiedź środkowa.

c) Zdający powinni dokonać konwersji liczby 1000_{16} na system dziesiętny i ósemkowy. Pierwsza odpowiedź jest nieprawidłowa, ponieważ w liczbie zapisanej w systemie piątkowym nie może pojawić się cyfra 5.

d) Firewall działa na zasadzie filtrowania pakietów, czyli sprawdzania pochodzenia pakietów i akceptowania pożądanych. To właśnie oprogramowanie tego typu zapobiega włamaniom do systemu. Keyloggery – przeciwnie – przejmują kontrolę nad procedurami systemu operacyjnego służącymi do obsługi klawiatury, umożliwiając ujawnienie wpisywanych haseł. Spam to niechciane lub niepotrzebne wiadomości elektroniczne. Filtry antyspamowe odrzucają e-maile pochodzące z podejrzanych źródeł oraz analizują zawartość wiadomości pod kątem zawierania spamu. Sam spam nie skutkuje właniem do systemu, dlatego ta odpowiedź również nie jest prawdziwa.

e) Wszystkie podane formaty są formatami służącymi do zapisu grafiki rastrowej. Format BMP, jest uniwersalny, popularny na wszystkich platformach systemowych. W formacie JPG zapisuje się grafikę publikowaną w Internecie, metody kompresji stratnej umożliwiają uzyskanie plików o niewielkich rozmiarach, kosztem utraty szczegółów na ogół niezauważalnych na ekranach monitorów. Format GIF stosuje się do zapisywania prostej (kolor 8-bitowy) grafiki umieszczanej na stronach WWW oraz animacji, umożliwia zapisywanie przezroczystości.

Cześć II

Zadanie 4. Anagram (10 pkt)

Anagram to słowo powstałe z innego słowa przez przestawienie liter. Przez słowo rozumiemy w tym zadaniu dowolny ciąg liter alfabetu łacińskiego.

Przykłady anagramów:

dla słowa: *barok* – *korba*, *robak*, *arobk*, *roka*, *orkab* ...

dla słowa: *ranty* – *tyran*, *narty*, *ntyra*, *natyr*, *ytar* ...

W pliku tekstowym *anagram.txt* znajduje się 200 wierszy zawierających po 5 słów w każdym wierszu. Słowa oddzielone są znakiem odstępu. Długość każdego ze słów wynosi od 1 do 20 znaków.

Przykład:

```
abcd cdba dbac cbad dcba
barbakan xle ala foto ofot
smok ayszk lamp ayszk bakara
skok arabanta oko agnieba dyskietka
.....
```

Napisz **program** w wybranym przez siebie języku programowania, za pomocą którego wykonasz poniższe polecenia:

f) Wyszukaj w pliku *anagram.txt* te wiersze, w których wszystkie słowa znajdujące się w danym wierszu mają taką samą liczbę znaków. Zapisz te wiersze w pliku *odp_4a.txt*.

- g) Wyszukaj w pliku `anagram.txt` wszystkie wiersze tekstu, w których wszystkie słowa są anagramami pierwszego słowa w danym wierszu. Zapisz te wiersze w pliku `odp_4b.txt`.

Do oceny oddajesz plik(i) o nazwie(ach),
 tu wpisz nazwę(y) pliku(ów)
 zawierający(e) komputerową(e) realizację(e) Twoich obliczeń, pliki tekstowe `odp_4a.txt`
 i `odp_4b.txt`.

Sprawdzane umiejętności

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

- projektuje i przeprowadza wszystkie etapy na drodze do otrzymania informatycznego rozwiązania problemu,
- wykorzystuje metody informatyki w rozwiązywaniu problemów

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
4.a	wyszukuje w pliku wiersze, w których wszystkie słowa mają taką samą liczbę znaków	4	0,28	0,30	-	0,22
4.b	wyszukuje w pliku wiersze tekstu, w których wszystkie słowa są anagramami pierwszego słowa w danym wierszu	6	0,15	0,15	-	0,19

Przykładowy poprawny zapis rozwiązania:

przykładowe rozwiązanie w C++:

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

using namespace std;

string sortowanie(string wyraz)
{
    char temp;
    int i,k;
    for(i=1;i<=wyraz.length();i++)
    {
        temp=wyraz[i];
        k=i-1;
        while (k>=0&&wyraz[k]>temp)
        {
            wyraz[k+1]=wyraz[k];
            k--;
        }
        wyraz[k+1]=temp;
    }
    return wyraz;
}

int main()
```

```
{
string w1,w2,w3,w4,w5;
ifstream we ("anagram.txt");
ofstream wy1 ("odp_4a.txt");
ofstream wy2 ("odp_4b.txt");

while (!we.eof())
{
we>>w1>>w2>>w3>>w4>>w5;

if((w1.length()==w2.length())&&(w1.length()==w3.length())&&
(w1.length()==w4.length())&&(w1.length()==w5.length()))
{
wy1<<w1<<" "<<w2<<" "<<w3<<" "<<w4<<" "<<w5<<endl;

if((sortowanie(w1)==sortowanie(w2))&&(sortowanie(w1)==sortowanie(w3))&&
(sortowanie(w1)==sortowanie(w4))&&(sortowanie(w1)==sortowanie(w5)))
wy2<<w1<<" "<<w2<<" "<<w3<<" "<<w4<<" "<<w5<<endl;
}
}

we.close();
wy1.close();
wy2.close();
system ("pause");

return 0;
}
```

przykładowe rozwiązanie w Pascalu

```
uses crt;
var
plo, plz : text;
t : array[1..5] of string;
i,j, linia : byte;
s : string;

function Anagramy(ss1,ss2:string): boolean;
var ii : byte;
Begin
Anagramy:= true;
if length(ss2) <> length(ss1) then
Anagramy:= false
else
begin
for ii:= 1 to length(ss1) do
if pos(ss1[ii],ss2) = 0 then
begin
Anagramy:= false;
exit;
end
else
delete(ss2,pos(ss1[ii],ss2),1);
end;
End;

BEGIN
clrscr;
assign(plo,'anagram.txt');
reset(plo);
```



```
assign(plz,'odp_4b.txt');
rewrite(plz);

while not eof(plo) do
begin
for j:= 1 to 5 do
t[j]:='';
readln(plo,s);
j:= 1;
for i:= 1 to length(s) do
if s[i] <> ' ' then
begin
t[j]:= t[j] + s[i];
end
else
inc(j);

if (Anagramy(t[1],t[2])) and (Anagramy(t[1],t[3])) and Anagramy(t[1],t[4]))
and (Anagramy(t[1],t[5]))
then
begin
writeln(plz,s);
writeln(s);
end;
end;

close(plo);
close(plz);
readkey;
END.
```

Komentarz:

Zadanie 4 miało charakter programistyczny. Wyróżniało się tym, iż w treści wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego. Podpunkt a) był stosunkowo prosty, głównym problemem było wczytanie danych z pliku (zestawu 200 wierszy po pięć słów oddzielonych znakiem spacji), porównanie długości słów i zapisanie wierszy w pliku odpowiedzi.

W podpunkcie b) należało już zapisać konkretny algorytm sprawdzający, czy słowa zapisane w wierszu są anagramami. Spotyka się najczęściej dwa rozwiązania tego problemu. Pierwszy polega na posortowaniu znaków we wszystkich słowach i sprawdzeniu, czy posortowane łańcuchy znaków są identyczne (rozwiązanie w C++). Drugie rozwiązanie polega na sprawdzaniu, czy kolejne znaki pierwszego słowa występują w słowie drugim, a następnie usunięciu już sprawdzonego znaku z drugiego słowa (rozwiązanie w Pascalu). Funkcją służącą do wyszukiwania i określania pozycji łańcucha B wewnątrz łańcucha A jest: w Pascalu $Pos(B,A)$, w C++ $A.find(B)$, w języku Java jest $A.indexOf(B)$. Ogólny wskaźnik łatwości dla tego zadania jest niepokojąco niski, odpowiednio 0,28 dla a) i 0,15 dla b), bo sam problem do rozwiązania jest bardzo prosty. Jedynej trudności upatrywać tu można w zapisie znanych uczniom prostych algorytmów na poprawnie działający program w konkretnym języku programowania. Okazało się, że maturzystom wybierającym poziom rozszerzony sprawiło to trudność. Tylko nieliczni rozwiązyli to zadanie prawidłowo.

Zadanie 5. Numer PESEL (10 pkt)

Numer PESEL to 11-cyfrowy kod, jednoznacznie identyfikujący określoną osobę fizyczną. Dla wszystkich urodzonych przed 1.1.2000 r. skonstruowany został w następujący sposób:

- cyfry od 1 do 6 to data urodzenia (w kolejności: rok, miesiąc, dzień),
- cyfry od 7 do 9 to liczba porządkowa,
- cyfra 10-ta to płeć (cyfra parzysta dla kobiet, nieparzysta dla mężczyzn),
- cyfra 11-ta to cyfra kontrolna.

Przykład:

Numer PESEL 75121968629 oznacza kobietę (bo przedostatnia cyfra w numerze jest parzysta), urodzoną 19 grudnia 1975 roku, której nadano liczbę porządkową 686.

Cyfra kontrolna jest obliczana wg następującego algorytmu:

1. Każdą cyfrę numeru PESEL (oprócz ostatniej, która jest cyfrą kontrolną) mnożymy przez odpowiadającą jej wagę. Tablica wag ma postać: **1, 3, 7, 9, 1, 3, 7, 9, 1, 3**.

W podanym powyżej numerze PESEL cyfrą kontrolną jest 9.

PESEL: 7 5 1 2 1 9 6 8 6 2 9 (ostatnią cyfrę pomijamy, bo to cyfra kontrolna)

WAGI: 1 3 7 9 1 3 7 9 1 3

WYNIKI: $7 \times 1 = 7$; $5 \times 3 = 15$; $1 \times 7 = 7$; $2 \times 9 = 18$; $1 \times 1 = 1$; $9 \times 3 = 27$; $6 \times 7 = 42$; $8 \times 9 = 72$; $6 \times 1 = 6$; $2 \times 3 = 6$

2. Wyniki sumujemy: $7 + 15 + 7 + 18 + 1 + 27 + 42 + 72 + 6 + 6 = 201$

3. Wyznaczamy resztę z dzielenia sumy przez 10: $201 : 10 = 20$ reszta = 1

4. Jeżeli reszta = 0, to cyfra kontrolna wynosi 0.

Jeżeli reszta $\neq 0$, to cyfra kontrolna będzie uzupełnieniem reszty do 10, czyli w podanym przykładzie jest to cyfra 9.

Plik `pesel.txt` zawiera numery PESEL 150 osób zatrudnionych w biurze obliczeniowym „Statystyk”. Korzystając z informacji zawartych w pliku `pesel.txt` oraz dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych podpunktów umieść w pliku `odp_5.txt`, poprzedzając je literą oznaczającą ten podpunkt.

Uwaga: Możesz przyjąć, że nawet gdy cyfra kontrolna numeru PESEL jest niepoprawna, to dane osobowe w nim zapisane są prawidłowe.

- a) Ile osób urodziło się w grudniu?
- b) Podaj, ile kobiet pracuje w biurze obliczeniowym.
- c) Podaj rok, w którym urodziło się najwięcej osób pracujących w biurze.
- d) Wyszukaj nieprawidłowe numery PESEL, w których nie zgadza się cyfra kontrolna. Posortuj te numery PESEL rosnąco.
- e) Sporządź zestawienie liczby osób urodzonych w kolejnych dziesięcioleciach, tzn. w latach pięćdziesiątych, sześćdziesiątych, siedemdziesiątych, osiemdziesiątych i dziewięćdziesiątych. Utwórz wykres ilustrujący procentowy rozkład liczby osób w poszczególnych przedziałach.

Uwaga: Do urodzonych w latach pięćdziesiątych należy zaliczyć osoby urodzone od roku 1950 do roku 1959 włącznie.

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
`odp_5.txt` zawierający wyniki.

Sprawdzane umiejętności

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

- dobiera właściwy program (użytkowy lub własnoręcznie napisany) do rozwiązywanego zadania

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.	wyszukuje w łańcuchu znaków (numerze PESEL) łańcucha '12' (osób urodzonych w grudniu).	1	0,84	0,86	-	0,72
2.	sprawdza, czy przedostatnia cyfra numeru PESEL jest parzysta.	1	0,69	0,72	-	0,61
3.	oblicza wartość najczęstszą.	2	0,76	0,77	-	0,72
4.	oblicza cyfrę kontrolną wg podanego algorytmu.	3	0,35	0,40	-	0,17
5.	sporządza zestawienie i wykres.	3	0,54	0,53	-	0,57

Przykładowy poprawny zapis rozwiązania:

Przykładowe rozwiązanie w arkuszu kalkulacyjnym Excel.

a) odp.: 20

=FRAGMENT.TEKSTU(A1;3;2)

=LICZ.JEŻELI(B1:B150;12)

b) odp.: 74

=FRAGMENT.TEKSTU(A1;10;1)

=MOD(D1;2)

=LICZ.JEŻELI(E1:E150;0)

c) odp.: 89 lub 1989

=FRAGMENT.TEKSTU(A1;1;2)

=WARTOŚĆ(G1)

=WYST.NAJCZĘŚCIEJ(H1:H150)

d) odp.:

54043010088

60061144469

77072919805

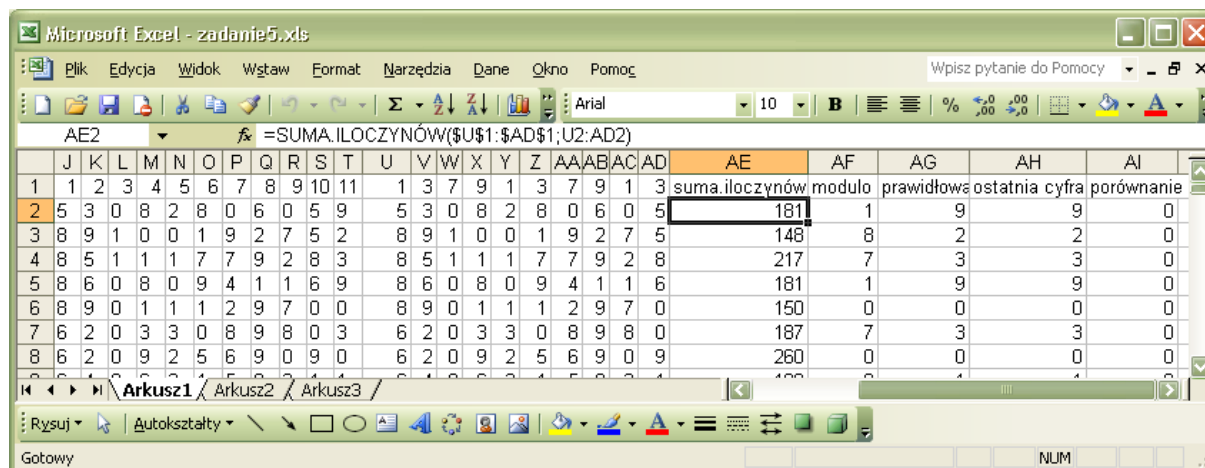
77120835871

83041812338

89081421445

91032272651

92022716243



	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	1	2	3	4	5	6	7	8	9	10	11	1	3	7	9	1	3	7	9	1	3	suma.iloczynów	modulo	prawidłowa	ostatnia cyfra	porównanie
2	5	3	0	8	2	8	0	6	0	5	9	5	3	0	8	2	8	0	6	0	5	181	1	9	9	0
3	8	9	1	0	0	1	9	2	7	5	2	8	9	1	0	0	1	9	2	7	5	148	8	2	2	0
4	8	5	1	1	1	7	7	9	2	8	3	8	5	1	1	1	7	7	9	2	8	217	7	3	3	0
5	8	6	0	8	0	9	4	1	1	6	9	8	6	0	8	0	9	4	1	1	6	181	1	9	9	0
6	8	9	0	1	1	1	2	9	7	0	0	8	9	0	1	1	1	2	9	7	0	150	0	0	0	0
7	6	2	0	3	3	0	8	9	8	0	3	6	2	0	3	3	0	8	9	8	0	187	7	3	3	0
8	6	2	0	9	2	5	6	9	0	9	0	6	2	0	9	2	5	6	9	0	9	260	0	0	0	0

Za pomocą funkcji FRAGMENT.TEKSTU zapisujemy w kolejnych kolumnach kolejne cyfry numeru PESEL, wartość tekstową przekształcamy w liczbę:

=FRAGMENT.TEKSTU(\$A2;J\$1;1)

=WARTOŚĆ(J2)

Zgodnie z algorytmem podanym w treści zadania tworzymy sumę iloczynów:

=SUMA.ILOCZYNÓW(\$U\$1:\$AD\$1;U2:AD2)

Wyznaczamy resztę z dzielenia sumy przez 10:

=MOD(AE2;10)

Obliczamy cyfrę kontrolną:

=JEŻELI(AF2=0;0;10-AF2)

Porównujemy prawidłową cyfrę kontrolną z cyframi numerów PESEL podanymi w zadaniu:

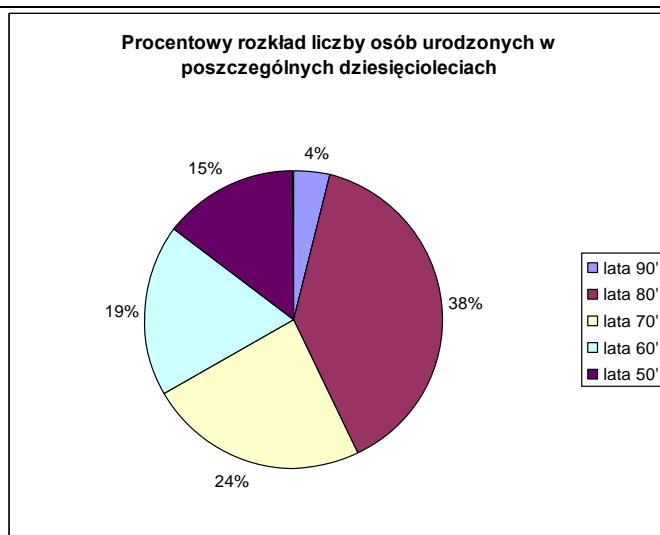
=JEŻELI(AG2=AH2;0;1)

Filtrujemy rekordy z wartością 1. Numery PESEL w tych rekordach są nieprawidłowe ze względu na cyfrę kontrolną.

e) zestawienie:

lata	liczba urodzonych
50-te	22
60-te	28
70-te	36
80-te	58
90-te	6

wykres:

**Komentarz:**

To zadanie maturzyści najczęściej rozwiązywali w arkuszu kalkulacyjnym, zdarzały się również rozwiązania programistyczne, szczególnie podpunktu d). Najprostszy okazał się podpunkt a): wystarczyło z numeru PESEL pobrać trzecią i czwartą cyfrę, a następnie policzyć, ile wśród pobranych numerów miesięcy występuje 12–stek. W podpunkcie b) wystarczyło policzyć liczbę numerów PESEL, w których przedostatnia cyfra jest parzysta, co prosto sprawdzić funkcją MOD. W podpunkcie c) po wyodrębnieniu roku urodzenia należało skorzystać ze znanej maturzystom funkcji WYST.NAJCZĘŚCIEJ. Podpunkt d) sprawił maturzystom największe trudności, prawidłowo rozwiązały go nieliczne osoby. Aby uzyskać dobry wynik w zadaniu, należało zastosować w odpowiedniej kolejności kilka funkcji, postępując zgodnie z algorytmem podanym w jego treści. Podpunkt e) okazał się umiarkowanie trudny, a rozwiązanie polegało na przygotowaniu standardowego zestawienia i wykresu procentowego do tego zestawienia.

Zadanie 6. Szkoła (10 pkt)

Szkoła dysponuje danymi zawartymi w trzech plikach: uczniowie.txt, oceny.txt, przedmioty.txt.

- Plik uczniowie.txt zawiera następujące dane o uczniach: *idUcznia*, *nazwisko*, *imie*, *ulica*, *dom*, *idKlasy*.
- Plik oceny.txt zawiera dane o ocenach: *idUcznia*, *ocena*, *data*, *idPrzedmiotu*.
- Plik przedmioty.txt zawiera dane o przedmiotach: *idPrzedmiotu*, *nazwaPrzedmiotu*, *nazwisko_naucz*, *imie_naucz*.

Korzystając z danych zawartych w plikach uczniowie.txt, oceny.txt, przedmioty.txt oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Każdą odpowiedź umieść w pliku odp_6.txt, poprzedzając ją oznaczeniem odpowiedniego podpunktu od a) do f).

- Poza rejonem szkoły leżą ulice Worcella oraz Sportowa. Podaj, ilu uczniów mieszka poza rejonem szkoły (czyli na jednej z tych dwóch ulic).
- Wypisz wszystkie oceny ucznia Jana Augustyniaka z języka polskiego.
- Oblicz, ile dziewcząt i ilu chłopców jest w poszczególnych klasach. Wynik przedstaw w postaci zestawienia: *idKlasy*, *liczba dziewcząt*, *liczba chłopców*. Załóż, że imiona dziewcząt (i tylko dziewcząt) kończą się na literę a.

- d) Utwórz zestawienie dla klasy 2a, zawierające nazwy przedmiotów i średnie ocen klasy z tych przedmiotów (średnie podaj z zaokrągleniem do dwóch miejsc po przecinku). Zestawienie posortuj nierosnąco według średnich ocen.
- e) Utwórz zestawienie uporządkowane alfabetycznie według nazwisk, zawierające wykaz osób z klasy 2c, które w kwietniu 2009 roku otrzymały oceny niedostateczne (imię, nazwisko, przedmiot).
- f) Podaj nazwisko, imię, klasę oraz średnią ocen osoby, która osiągnęła najwyższą średnią ocen w całej szkole (jest tylko jedna taka osoba).

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
 odp_6.txt zawierający wyniki.

Sprawdzane umiejętności

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

- projektuje relacyjne bazy danych i proste aplikacje bazodanowe,
- stosuje metody wyszukiwania i przetwarzania informacji w relacyjnych bazach danych.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.	projektuje relacyjną bazę danych, wyszukuje informacje	1	0,83	0,85	-	0,72
2.	projektuje relacyjną bazę danych, wyszukuje informacje	1	0,72	0,77	-	0,50
3.	projektuje relacyjną bazę danych, wyszukuje informacje	2	0,33	0,34	-	0,31
4.	projektuje relacyjną bazę danych, wyszukuje informacje, porządkuje informacje, zapisuje w zadanym formacie	2	0,34	0,41	-	0,08
5.	projektuje relacyjną bazę danych, wyszukuje informacje, porządkuje informacje,	2	0,33	0,34	-	0,28
6.	projektuje relacyjną bazę danych, wyszukuje informacje,	2	0,37	0,41	-	0,19

Przykładowy poprawny zapis rozwiązania:

a) odp.: 15

```
SELECT Count(*) AS [Ilosc uczniow] FROM Uczniowie WHERE
(((Uczniowie.ulica)="Worcella" Or Uczniowie.ulica)="Sportowa"));
```

b) odp.: 3,1,1

```
SELECT Uczniowie.imie, Uczniowie.nazwisko,
Przedmioty.NazwaPrzedmiotu, Oceny.Ocena FROM (Oceny INNER JOIN
Uczniowie ON Oceny.IDucznia=Uczniowie.IDucznia) INNER JOIN
Przedmioty ON Oceny.IDprzedmiotu=Przedmioty.IDprzedmiotu
WHERE (((Uczniowie.imie) Like "Jan") AND ((Uczniowie.nazwisko)
Like "Augustyniak") AND ((Przedmioty.NazwaPrzedmiotu) Like
"Polski"));
```

h) odp.:

IDklasy	Liczba dziewcząt	Liczba chłopców
1a	26	4
1b	26	3
1c	28	2
1d	26	1
1e	30	2
2a	30	1
2b	25	4
2c	27	1
2d	31	2
2e	19	2
3a	24	1
3b	24	1
3c	23	2
3d	19	6
3e	22	2

kwerenda pomocnicza C1

```
SELECT Uczniowie.IDklasy, Count(Uczniowie.IDucznia) AS Dziewcząt
FROM Uczniowie WHERE (((Uczniowie.imie) Like "*A")) GROUP BY
Uczniowie.IDklasy;
```

kwerenda pomocnicza C2

```
SELECT Uczniowie.IDklasy, Count(Uczniowie.IDucznia) AS Chłopców
FROM Uczniowie WHERE (((Uczniowie.imie) Not Like "*A")) GROUP BY
Uczniowie.IDklasy;
```

```
SELECT C1.IDklasy, C2.Chłopców, C1.Dziewcząt FROM C1 INNER JOIN
C2 ON C1.IDklasy = C2.IDklasy GROUP BY C1.IDklasy, C2.Chłopców,
C1.Dziewcząt;
```

i) odp.:

geografia	3,25
wf	3,14
niemiecki	3,12
angielski	3,11
chemia	3,03
historia	3,02
wos	3,01
fizyka	3,01
polski	2,95
matematyka	2,93
informatyka	2,72

```
SELECT Przedmioty.NazwaPrzedmiotu, Avg(Oceny.Ocena) AS
ŚredniaOfOcena FROM Uczniowie INNER JOIN (Oceny INNER JOIN
Przedmioty ON Oceny.IDprzedmiotu = Przedmioty.IDprzedmiotu) ON
Uczniowie.IDucznia = Oceny.IDucznia GROUP BY Uczniowie.IDklasy,
Przedmioty.NazwaPrzedmiotu HAVING (((Uczniowie.IDklasy) Like
"2a")) ORDER BY Avg(TOceny.Ocena) DESC;
```

j) odp.:

Urszula Bednarczyk wos
 Krystyna Lysek fizyka
 Barbara Mikołajczyk wos
 Grażyna Witczak angielski
 Dorota Wojcik wos
 Beata Zabelana niemiecki
 Beata Zabelana angielski

```
SELECT Uczniowie.IDklasy, Uczniowie.imie, Uczniowie.nazwisko,
Przedmioty.NazwaPrzedmiotu, Oceny.Ocena, Oceny.Data FROM
Uczniowie INNER JOIN (Oceny INNER JOIN Przedmioty ON
Oceny.IDprzedmiotu=Przedmioty.IDprzedmiotu) ON
Uczniowie.IDucznia=Oceny.IDucznia WHERE (((Uczniowie.IDklasy)
Like "2c") AND ((Oceny.Ocena) Like "1") AND ((Oceny.Data) Like
"2009-04-*")) ORDER BY Uczniowie.nazwisko;
```

k) odp.:

Barbara Ogozewska 2d 3,6

```
SELECT TOP 1 Uczniowie.imie, Uczniowie.nazwisko,
Uczniowie.IDklasy, Avg(Oceny.Ocena) AS ŚredniaOfOcena FROM
Uczniowie INNER JOIN (Przedmioty INNER JOIN Oceny ON
Przedmioty.IDprzedmiotu = Oceny.IDprzedmiotu) ON
Uczniowie.IDucznia = Oceny.IDucznia GROUP BY Uczniowie.imie,
Uczniowie.nazwisko, Uczniowie.IDklasy ORDER BY Avg(oceny.Ocena)
DESC;
```

Komentarz:

Ostatnie zadanie drugiego arkusza miało charakter bazodanowy. Dane do zadania umieszczono w trzech plikach opisujących uczniów, przedmioty i oceny. Pliki te odpowiadały trzem tabelom relacyjnej bazy danych, powiązanych poprzez identyfikator ucznia i identyfikator przedmiotu. 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 uczniami mieszkającymi na ulicach Worcella i Sportowej, a następnie policzyć ilość tych rekordów. W podpunkcie b) należało zbudować prostą kwerendę wybierającą, która poda oceny z j. polskiego jednego z uczniów. Podpunkty c), d), e) i f) okazały się trudne dla zdających. Rozwiązanie wymagało łączenia tabel, sortowania, filtrowania i grupowania danych. W podpunkcie d) pojawiały się błędy zaokrągleń, w e) występowały pomyłki (prawdopodobnie przez nieuwagę) – zdający podawali jedynie nazwiska bez przedmiotów lub zapominali o sortowaniu.

Dużo rzadziej niż w latach ubiegłych zdarzało się, że uczniowie zamieszczali tylko pliki tekstowe z odpowiedziami, nie dołączając plików realizacji obliczeń. Częściej pojawiała się sytuacja odwrotna – brak pliku tekstowego z odpowiedziami lub brak odpowiedzi w pliku tekstowym do niektórych punktów – egzaminatorzy znajdowali prawidłowe wyniki, przeglądając utworzone kwerendy lub pliki arkusza kalkulacyjnego.

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. Można je było rozwiązać różnymi metodami, przy pomocy różnych narzędzi. Decyzja wyboru metody i programu należała do zdającego. Zadania sformułowane były precyzyjnie i krótko. Poruszane problemy nie wykraczały poza standardy wymagań egzaminacyjnych wynikające z podstawy programowej. Ponadto z wypowiedzi zdających można wnioskować, że prawidłowo dostosowano czas przeznaczony na rozwiązanie zadań do czasu trwania egzaminu. Wyniki uzyskane przez maturzystów były różnorodne, co wskazuje na duże zróżnicowanie poziomu ich wiedzy. Dosyć istotne różnice w wynikach występowały pomiędzy typami szkół (liceum ogólnokształcące i technikum), a także pomiędzy konkretnymi szkołami, które kończyli maturzyści. W niektórych szkołach zdający doskonale radzili sobie z zadaniem, w innych wyniki były znacząco niższe.

Jak poprawić wyniki osiągane przez maturzystów i w jaki sposób ukierunkować przygotowania do przyszłorocznej matury?

W dużej mierze wyniki osiągane przez uczniów na egzaminie zależą od zastosowania przez nauczyciela przedmiotu informatyka 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 pomóc w ich przygotowaniu do egzaminu maturalnego. Są to:

- informator maturalny (zawiera opis struktury i formy egzaminu oraz wymagania egzaminacyjne),
- komentarz do zadań z przedmiotów matematyczno-przyrodniczych wydawany przez OKE lub CKE (zawiera przykładowe rozwiązania zadań, najczęstsze błędy i ich przyczyny, które należy eliminować w procesie nauczania)
- arkusze egzaminacyjne z lat poprzednich, udostępnione na stronie CKE, OKE.

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.). Zdający 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.

W pierwszej połowie lipca 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, umiejętność korzystania z systemu pomocy mogą okazać się niezwykle przydatne podczas egzaminu, kiedy uczeń ma limitowany czas na rozwiązanie zadań i szkoda go poświęcać na zapoznavanie się ze środowiskiem.