

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,
- *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. Zegar (5 pkt)

Na jednej z uczelni informatycznych nad wejściem do auli umieszczony został elektroniczny zegar odliczający sekundy od rozpoczęcia wykładu do jego zakończenia. Zegar jest nietypowy, ponieważ liczba sekund, która upływa od rozpoczęcia wykładu wyświetlana jest w systemie o podstawie 2.

Przed rozpoczęciem odliczania zegar jest wyzerowany, tzn. na pierwszym polu od prawej strony jest wyświetlane zero i pozostałe pola są wygaszone.

Przykład:

Po upływie 10 sekund na zegarze wyświetlone są 4 pola z napisem: 1010.

Po upływie 25 sekund – 5 pól z napisem: 11001.

Wykonaj następujące polecenia:

- a) Oblicz, na ilu polach tego zegara będzie wyświetlony czas najdłuższego wykładu, który może trwać 240 minut, czyli 14400 sekund.

[illegible]

Sprawdzane umiejętności

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

- zna sposoby reprezentacji w komputerze liczb,
- zna podstawowe algorytmy i techniki algorytmiczne (zna pozycyjne reprezentacje liczb),
- stosuje podstawowe algorytmy i struktury danych w rozwiązywaniu problemów informatycznych.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.a.	sprawdza długość zapisu binarnego wykładu trwającego 14400 sekund.	1	0,79	0,81	–	0,78
1.b.	zamienia liczbę zapisaną w systemie binarnym na liczbę w systemie dziesiętnym.	1	0,67	0,74	–	0,64
1.c.	zapisuje algorytm, który oblicza liczbę jedynek w zapisie binarnym podanej liczby dziesiętnej.	3	0,37	0,52	–	0,29

Przykładowy poprawny zapis rozwiązania:

1a.

$$14400_{10} = 11100001000000_2$$

Odp.:14

1b.

$$1111110100100_2 = 810_{10}$$

$$810:60=135$$

Odp.:135 min

1c.

rozwiązanie 1

$l \leftarrow 0$

dopóki $s > 0$

jeśli $(s \bmod 2 = 1)$ to $l \leftarrow l+1$

$s \leftarrow s \div 2$

wypisz l

rozwiązanie 2

$l \leftarrow 0, p \leftarrow 0$

dopóki $p < s$

$p \leftarrow p * 2$

dopóki $s > 0$

jeśli $p \leq s$

$l \leftarrow l+1$

$s \leftarrow s - p$

$p \leftarrow p \div 2$

wypisz l

Komentarz:

Zadanie pierwsze poświęcone było w całości zagadnieniom związanym z systemem binarnym. Składa się ono z trzech podpunktów, w tym pierwsze dwa można rozwiązać poprzez wykonanie zamiany reprezentacji konkretnych liczb z systemu dziesiętnego na binarny (punkt a) i odwrotnie (punkt b). Teoretycznie do rozwiązania punktu a) nie była konieczna zmiana reprezentacji liczby „a”, jedynie znajomość formuły określającej długość binarnej reprezentacji w zależności od wielkości liczby, czyli $\lceil \log_2(n+1) \rceil$. Jednak wydaje się, że dla uczniów bezpieczniejszym rozwiązaniem było wykonanie odpowiedniej konwersji między systemami.

W punkcie c) zadaniem maturzysty było samodzielne zapisanie algorytmu podającego liczbę jedynek w binarnym zapisie liczby podanej na wejściu. Była to najtrudniejsza część zadania, co znajduje odzwierciedlenie w punktacji (3 punkty, wobec 1 punktu za każdą z części a i b). Realizacja tego polecenia wymaga umiejętnego wykorzystania algorytmu konwersji liczby z systemu dziesiętnego na binarny. Maturzysta nie musiał się jednak martwić niewłaściwą kolejnością cyfr uzyskiwanych kolejno w algorytmie konwersji (od najmniej do najbardziej znaczącej). Z drugiej strony, realizacja polecenia wymagała umiejętności stosowania liczników w podstawowych konstrukcjach algorytmicznych.

Wskaźnik łatwości 0,52 klasyfikuje to zadanie jako umiarkowanie trudne. O ile konwersja liczby dziesiętnej na system binarny oraz operacja odwrotna nie sprawiły zdającym większych problemów, to w przypadku zapisania algorytmu pojawiły się kłopoty. Niektórzy nie zwrócili uwagi na specyfikację i sumowali jedynki bez konwersji liczby s na liczbę binarną, ponadto występowały błędy w organizacji pętli sterującej. Uczniowie wybierali często zapis w wybranym języku programowania. Obok standardowych rozwiązań pojawiały się nieliczne rozwiązania rekurencyjne, a także ciekawe rozwiązania poprzez odejmowanie potęg dwójki (rozwiązanie 2).

Zadanie 2. Algorytm (8 pkt)

Poniżej przedstawiony został algorytm wypisujący dla zadanej liczby całkowitej $n \geq 2$ komunikat **TAK** lub **NIE**.

Algorytm:

krok 1. $i \leftarrow 2$

krok 2. jeśli $i \geq n$, wypisz **TAK** i przejdź do kroku 5

krok 3. jeśli $(n \bmod i) = 0$, wypisz **NIE** i przejdź do kroku 5

krok 4. $i \leftarrow i+1$ i przejdź do kroku 2

krok 5. zakończ wykonywanie algorytmu.

Uwaga: „ $n \bmod i$ ” oznacza resztę z dzielenia całkowitego liczby n przez i

Wykonaj następujące polecenia:

a) Przeanalizuj działanie algorytmu dla podanych wartości n i uzupełnij tabelę:

Wartość n	25	37	41	49
Wypisany komunikat				

b) Zaznacz znakiem **X** w odpowiedniej kolumnie tabeli, które zdania są prawdziwe (**P**), a które fałszywe (**F**).

		P	F
1.	Instrukcja warunkowa zapisana w kroku 3 wykona się dla każdego n dokładnie n razy.		

	wykonywanych w algorytmie operacji.					
2.c.	Zapisuje algorytm wypisujący wszystkie czynniki pierwsze podanej liczby.	4	0,43	0,58	-	0,34

Przykładowy poprawny zapis rozwiązania:

2.a.

prawidłowe komunikaty: *NIE, TAK, TAK, NIE*

2.b.

prawidłowe zaznaczenia: FPPF

2.c.

1. dzielnik $\leftarrow 2$
2. dopóki $n > 1$ wykonuj
 - dopóki $(n \bmod \text{dzielnik} = 0)$ wykonuj
 - wypisz dzielnik
 - $n \leftarrow n / \text{dzielnik}$
 - dzielnik $\leftarrow \text{dzielnik} + 1$

Komentarz:

Drugie zadanie arkusza I sprawdzało umiejętność śledzenia i analizy algorytmów zapisanych w pseudokodzie, a także, podobnie do pierwszego, wymaga samodzielnego sformułowania algorytmu. W treści zadania podany był algorytm sprawdzający pierwszość liczby podanej na wejściu. W punkcie a) należało prześledzić działanie algorytmu na czterech konkretnych liczbach i podać wypisywany komunikat. Następnie, w punkcie b), maturzysta odpowiadał na cztery pytania sprawdzające, czy zrozumiał on działanie algorytmu oraz potrafi przeanalizować czas jego działania. Podobnie do zadania pierwszego, ostatni podpunkt jest najtrudniejszy – wymaga samodzielnego sformułowania i zapisania algorytmu podającego rozkład liczby na czynniki pierwsze. Należy zaznaczyć, że zadanie to nie powinno nastręczać dużych trudności, choćby dlatego, że rozkład na czynniki pierwsze to jedno z klasycznych zadań w podstawowym kursie algorytmiki i programowania, a jego różne warianty już pojawiały się w zestawach maturalnych.

Prawie połowa rozwiązań zapisana była w języku programowania, który uczeń wybrał na egzamin. Najczęstsze błędy pojawiały się w organizacji pętli sterującej (błąd w warunku lub nieprawidłowe działanie licznika pętli) oraz nieprawidłowe obniżanie wartości liczby (dzielenie całkowite).

Zadanie 3. Test (7 pkt)

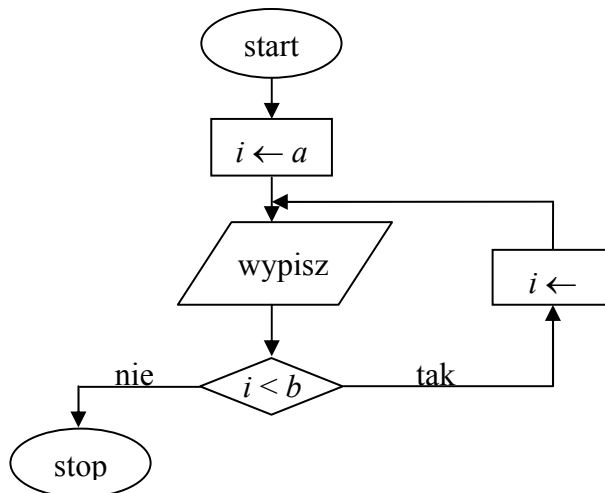
Dla następujących zdań **zaznacz znakiem X** właściwe odpowiedzi.

Uwaga: W każdym podpunkcie poprawna jest tylko jedna odpowiedź.

a) Dane są dwie liczby: $A=11001_2$ oraz $B=1010_2$

- ☐ $A \cdot B = 101_{10}$.
- ☐ $A + B = 35_{10}$.
- ☐ $A - B = 10100_2$.

b) Dane są dwie liczby całkowite a, b takie, że $b > a$ oraz schemat algorytmu:



Wynikiem działania tego algorytmu jest wypisanie

- ☐ wyłącznie liczb parzystych z przedziału domkniętego $\langle a, b \rangle$.
- ☐ wyłącznie liczb nieparzystych z przedziału domkniętego $\langle a, b \rangle$.
- ☐ wszystkich liczb z przedziału domkniętego $\langle a, b \rangle$.

c) Do odbierania wiadomości za pomocą poczty elektronicznej służy protokół

- ☐ SMTP.
- ☐ POP3.
- ☐ FTP.

d) W sieciach komputerowych

- ☐ komputery pracujące w tej samej sieci mają identyczne adresy IP.
- ☐ LAN to komputer podłączony do sieci, posiadający adres IP.
- ☐ serwer DNS tłumaczy adresy domenowe na adresy IP.

e) Firewall to program

- ☐ umożliwiający filtrowanie pakietów danych.
- ☐ służący do sprawdzania błędów na dysku.
- ☐ zwiększający przepustowość w sieciach komputerowych.

f) W relacyjnym modelu bazy danych

- ☐ wyróżniamy trzy rodzaje relacji.
- ☐ podstawową formą organizacji danych są kwerendy.
- ☐ kluczem podstawowym nie może być kolumna zawierająca tylko i wyłącznie teksty.

g) Programowanie strukturalne polega między innymi na

- ☐ budowaniu hierarchicznej struktury folderów.
- ☐ hierarchicznym podziale programu na wyodrębnione podzadania.
- ☐ hierarchicznym podziale długiego tekstu.

Sprawdzane umiejętności

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

- zna sposoby reprezentowania w komputerze liczb,
- zna zasady konwersji pomiędzy systemem binarnym i dziesiętnym,

- zna pojęcie algorytmu i różne sposoby jego zapisu,
- opisuje usługi oferowane w sieciach komputerowych,
- przedstawia budowę i funkcjonowanie komputerowej sieci lokalnej,
- zna i omawia typowe narzędzia służące do zabezpieczania programów i danych w komputerze,
- zna i omawia podstawowe formy organizacji informacji w bazach danych,
- zna zasady programowania strukturalnego.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a.	wykonuje działania arytmetyczne na liczbach zapisanych w systemie binarnym.	1	0,81	0,77	–	0,84
3.b.	analizuje algorytm zapisany w postaci schematu blokowego.	1	0,94	0,91	–	0,96
3.c.	rozpoznaje protokół pocztowy.	1	0,79	0,72	–	0,82
3.d.	rozpoznaje pojęcia związane z architekturą sieciową i adresacją IP.	1	0,92	0,91	–	0,93
3.e.	zna działanie programów typu Firewall.	1	0,97	0,94	–	0,99
3.f.	zna podstawowe formy organizacji informacji w bazach danych.	1	0,52	0,45	–	0,56
3.g.	zna pojęcia związane z programowaniem strukturalnym.	1	0,93	0,92	–	0,94

Poprawny zapis rozwiązania:

a2, b3, c2, d3, e1, f1, g2

Komentarz:

Zadanie miało charakter testu wyboru sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Podpunkty od a) do g) z pominięciem f) okazały się dla zdających łatwe i bardzo łatwe, natomiast f) – umiarkowanie trudny.

Poziom podstawowy

Arkusz II

Zadanie 4. Hasła (10 pkt)

Informatyk z firmy „KompOK” zapisał w pliku `hasla.txt` 200 haseł. Każde hasło umieszczone jest w osobnym wierszu pliku. Hasło składa się tylko z małych liter alfabetu angielskiego, zaś jego długość wynosi od 3 do 10 znaków.

Wykorzystując dane zawarte w tym pliku, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych podpunktów zapisz w plikach tekstowych o nazwach `wynik4a.txt`, `wynik4b.txt`, `wynik4c.txt`.

W pliku `wynik4a.txt` podaj, ile haseł ma parzystą, a ile nieparzystą liczbę znaków.

W pliku wynik4b.txt utwórz zestawienie haseł (po jednym w wierszu), które są palindromami.

Palindrom to wyraz brzmiący tak samo przy czytaniu z lewej strony do prawej, jak i odwrotnie, np. *kajak*, *potop*.

Zapisz w pliku wynik4c.txt zestawienie haseł (po jednym w wierszu) zawierających w sobie dwa kolejne znaki, których suma kodów ASCII wynosi 220.

Przykłady:

Hasło *krzysio* zawiera dwa kolejne znaki *si*, których suma kodów ASCII wynosi 220. Kod ASCII znaku *s* to 115, kod znaku *i* to 105; suma kodów wynosi $115+105 = 220$.

Hasło *cyrk* zawiera również takie dwa kolejne znaki. Kod ASCII znaku *c* to 99, kod ASCII znaku *y* to 121; suma kodów wynosi $99+121=220$

Tabela kodów ASCII

Znak	a	b	c	d	e	f	g	h	i	j	k	l	m
Kod ASCII	97	98	99	100	101	102	103	104	105	106	107	108	109
Znak	n	o	p	q	r	s	t	u	v	w	x	y	z
Kod ASCII	110	111	112	113	114	115	116	117	118	119	120	121	122

Uwaga: Kolejność haseł w plikach wynik4b.txt, wynik4c.txt powinna być zgodna z kolejnością ich występowania w pliku hasla.txt.

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

zawierający(e) komputerową(e) realizację(e) Twojego rozwiązania do wszystkich podpunktów, plik tekstowy o nazwie wynik4a.txt, zawierający odpowiedź do podpunktu a), plik tekstowy o nazwie wynik4b.txt, zawierający wyniki z podpunktu b), plik tekstowy o nazwie wynik4c.txt, zawierający wyniki z podpunktu c).

Sprawdzane umiejętności

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

- tworzy specyfikację problemu, proponuje i analizuje jego rozwiązanie,
- formułuje informatyczne rozwiązanie problemu przez dobór algorytmu oraz odpowiednich struktur danych i realizuje je w wybranym języku programowania,
- posługuje się kompilatorem wybranego języka programowania.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
4.a	pisze program sprawdzający parzystą i nieparzystą liczbę znaków w podanych hasłach.	2	0,35	0,34	–	0,36
4.b	pisze program, sprawdzający które z podanych haseł są palindromami.	4	0,19	0,21	–	0,19
4.c	pisze program, sprawdzający które z haseł zawiera w sobie dwa	4	0,14	0,11	–	0,16

	kolejne znaki, których suma kodów ASCII wynosi 220.					
Przykładowy poprawny zapis rozwiązania: odp.: 4.a. parzyste: 105 nieparzyste: 95 4.b. Palindromy: dompmo grafarg kajak komok matam mpoopm mpouiuopm oddo omo pokop plkjklp pokkop plolp 4.c.						
amodda damod damodd dompmo edamo edamod isksad iughd kisksa kkompo komok kompiel kompo kompoc kompok kompoot kompooto kompost kompot komput komu komunikat moddam nruiugh nruiughd okisks		omnibus omo ompioroip ompoci ompokk ompooto ompootoo ruiughd ruiughdf sokisk sunruiug ughdf ughdfbk uiughdf uiughdfb unruiug unruiugh zedamo plkjklp mops polewa komputer komputererek kolomp plomp plolp				

przykładowe rozwiązanie w języku Pascal:

```
program zad4PP_2011;
uses crt;
var k,m,i:integer;
    tajne:boolean;
    plik1, plik2, plik3, plik4:text;
    x:string;

function palind(var a:string):boolean;
var i,j,k:integer;
begin
    j:=length(a);
    k:=j div 2;
    i:=1;
    palind:=true;
    while ((i<=k) and (palind)) do
    begin
        if a[i]<>a[j] then palind:=false;
        i:=i+1;
        j:=j-1;
    end;
end;

begin
    clrscr;

    k:=0;
    m:=0;
    assign(plik1,'hasla.txt');
    assign(plik4,'wynik4a.txt');
    assign(plik2,'wynik4b.txt');
    assign(plik3,'wynik4c.txt');
    rewrite(plik4);
    rewrite(plik3);
    rewrite(plik2);
    reset(plik1);
    repeat
        readln(plik1,x);
        if length(x)mod 2=0 then k:=k+1 else m:=m+1;
        if palind(x) then writeln(plik2,x);
        tajne:=false;
        i:=0;
        repeat
            i:=i+1;
            if ord(x[i])+ord(x[i+1])=220 then tajne:=true;
        until (i=length(x)-1) or (tajne=true);
        if tajne then writeln(plik3,x)
    until eof(plik1);
    writeln(plik4,'parzyste: ',k,' nieparzyste: ',m);
```

```

    close(plik1);
    close(plik2);
    close(plik3);
    close(plik4);
end.

```

przykładowe rozwiązanie w C++:

```

#include<iostream>
#include<fstream>
using namespace std;

int i=0,p=0,np=0,len=0,j=0,czy=0,dl=0;
string haslo[200],slovo;

ifstream odczyt("hasla.txt");
ofstream odp1("wynik4a.txt");
ofstream odp2("wynik4b.txt");
ofstream odp3("wynik4c.txt");

int main()
{
    while (!odczyt.eof()) // odczyt danych z pliku
    {
        odczyt >> haslo[i];
        i++;
    }
    //Odpowiedz na pytanie a
    for (int i=0; i<200; i++)
    {
        if (haslo[i].size()%2==0)
            {p++;} else {np++;} //parzystość
    }

    odp1 << p << " hasel ma parzysta ilosc znakow" << endl;
    odp1 << np << " hasel ma NIEparzysta ilosc znakow" <<
endl;

    //Odpowiedz na pytanie b
    for (int i=0; i<200; i++)
    {
        slovo = haslo[i];
        len = slovo.size();
        dl = len/2;
        czy=0;
        for (j=0; j<(len/2); j++)
        {
            if (slovo[j]==slovo[len-j-1]) { czy++; }
        }

        if (czy==dl) { odp2 << slovo << endl;}
    }
}

```

```

    }

    //Odpowiedz na pytanie c
    for (int i=0; i<200; i++)
    {
        slowo = haslo[i];
        len = slowo.size();
        dl = len/2;
        czy=0;
        for (int j=0; j<len; j++)
        {
            if ((int(slowo[j])+int(slowo[j+1]))==220)
            {czy++;}
        }

        if (czy!=0) {odp3 << slowo << endl;}
    }

    system("PAUSE");
    return 0;
}

```

Komentarz:

Zadanie czwarte miało charakter programistyczny. Wprawdzie w treści nie wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego, ale było to najbardziej optymalne rozwiązanie. Osoby, które nie posiadały wystarczającej biegłości w programowaniu, próbowały uzyskać odpowiedzi, korzystając z arkusza kalkulacyjnego. O ile nie było problemu z tym w podpunkcie a), to dla pozostałych podpunktów taki sposób był pracochłonny i często uzyskiwano odpowiedzi z błędami.

W podpunkcie b) 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 na tekstach.

Dla rozwiązania punktu c) konieczna była konwersja między znakami a ich kodami ASCII. Po wykonaniu konwersji należało sprawdzić, czy w słowie występują dwie sąsiednie wartości sumujące się do 220.

Zadanie 4c miało najwyższą frakcję opuszczeń spośród wszystkich zadań w arkuszu. Wśród rozwiązań pojawiały się błędy związane z dwukrotnym zliczaniem jednego słowa, co związane było z brakiem przerwania (break) działania pętli w przypadku znalezienia pierwszego stringa dwuliterowego, którego suma kodów ASCII wynosiła 220.

Zadanie 5. Domki (10 pkt)

Ośrodek wypoczynkowy „Promyk” wynajmuje domki letniskowe pracownikom. Ze względu na dużą liczbę chętnych nałożono ograniczenie – **pracownik może zarezerwować domek tylko raz w ciągu roku**. Dane są trzy pliki tekstowe o nazwach: domki.txt, pracownicy.txt, rezerwacje.txt. Zawierają one informacje na temat domków, pracowników i rezerwacji domków wykonanych przez pracowników w 2010 roku.

Dane w wierszach każdego z plików rozdzielone są pojedynczymi znakami odstępu, pierwszy wiersz każdego pliku jest wierszem nagłówkowym.

Plik domki.txt zawiera następujące dane: numer domku (*NrDomku*), liczbę pokoi (*LiczbaPokoi*), dostępność garażu (*Garaz*) oraz cenę za dobę (*CenaZaDobe*).

Przykład:

```
NrDomku LiczbaPokoi Garaz CenaZaDobe
1 4 Tak 200
2 4 Nie 160
```

Plik pracownicy.txt zawiera następujące dane: identyfikator pracownika (*IDpracownika*), nazwisko (*Nazwisko*) i imię (*Imie*).

Przykład:

```
IDpracownika Nazwisko Imie
1 Wroblewski Jan
2 Wiecek Jaremi
```

Plik rezerwacje.txt zawiera dane o rezerwacjach: numer rezerwacji (*NrRezerwacji*), identyfikator pracownika (*IdPracownika*), numer rezerwowanego domku (*NrDomku*) oraz liczbę zarezerwowanych dni (*LiczbaDni*).

Przykład:

```
NrRezerwacji IdPracownika NrDomku LiczbaDni
1 5 2 2
2 20 5 2
```

Korzystając z danych zawartych w plikach domki.txt, pracownicy.txt i rezerwacje.txt oraz z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych podpunktów umieść w pliku wyniki5.txt, poprzedzając je literami oznaczającymi te podpunkty.

- Utwórz zestawienie zawierające dla każdego domku jego numer oraz łączną liczbę dni, na które ten domek był zarezerwowany w ciągu całego sezonu.
- Podaj nazwiska i imiona pracowników, którzy rezerwowali domek nr 2. Zestawienie posortuj alfabetycznie według imion pracowników.
- Podaj nazwisko i imię pracownika, który zapłacił najwięcej za wynajem domku oraz kwotę, którą zapłacił (zgodnie z rezerwacją). Jest tylko jeden taki pracownik.
- Podaj liczbę rezerwacji domków z garażem oraz liczbę rezerwacji domków bez garażu.
- Utwórz zestawienie najdłuższych rezerwacji dla poszczególnych domków. W zestawieniu podaj dla każdego domku jego numer oraz liczbę dni najdłuższej rezerwacji dla tego domku.

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 i III obszaru standardów. Zdający:

- 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
5.a.	grupuje i sumuje dane.	2	0,45	0,42	–	0,46
5.b.	filtruje i sortuje dane.	2	0,55	0,70	–	0,47

5.c.	tworzy wyrażenie i wyszukuje maksimum.	2	0,42	0,57	–	0,34
5.d.	wykorzystuje grupowanie danych i funkcje agregujące.	2	0,49	0,60	–	0,43
5.e.	wykorzystuje grupowanie danych i funkcje agregujące.	2	0,41	0,47	–	0,37

Przykładowy poprawny zapis rozwiązania:

a) Odp.:

Nr Domku	Liczba Dni
1	29
2	58
3	51
4	39
5	39
6	35
7	44
8	48
9	25

```
SELECT Rezerwacje.NrDomku, Sum(Rezerwacje.LiczbaDni) AS
SumaOfLiczbaDni
FROM Rezerwacje GROUP BY Rezerwacje.NrDomku;
```

b) Odp.:

Nazwisko	Imie
Adamski	Adam
Sadej	Andrzej
Hamerak	Blazej
Bardzewska	Grazyna
Magdzinska	Irena
Wawrzynowski	Marek
Haberko	Marek
Piekarzewski	Pawel
Kowalski	Sebastian
Iwaszkiewicz	Slawomir
Zawada	Witold
Piasecki	Zbigniew
Lehmann	Zdzislaw
Kasprzak	Zofia

```
SELECT Pracownicy.Imie, Pracownicy.Nazwisko, Rezerwacje.NrDomku
FROM Pracownicy INNER JOIN Rezerwacje ON
Pracownicy.IDpracownika = Rezerwacje.IdPracownika WHERE
((Rezerwacje.NrDomku)=2));
```

c) Odp.:

Tadeusz Fickowski, 2500 zł

```
SELECT Pracownicy.Nazwisko, Pracownicy.Imie,
Rezerwacje![LiczbaDni]*[Domki]![CenaZaDobe] AS Wyr1 FROM Domki
INNER JOIN (Pracownicy INNER JOIN Rezerwacje ON
```

```
Pracownicy.IDpracownika = Rezerwacje.IdPracownika) ON  
Domki.NrDomku = Rezerwacje.NrDomku;
```

d) Odp.:

rezerwacja domku z garażem – 49

rezerwacja domku bez garażu – 48

```
SELECT Count(Rezerwacje.NrRezerwacji) AS PoliczOfNrRezerwacji,  
Domki.Garaz FROM Domki INNER JOIN Rezerwacje ON Domki.NrDomku =  
Rezerwacje.NrDomku GROUP BY Domki.Garaz HAVING (((Domki.Garaz)  
Like 'TAK'));
```

```
SELECT Count(Rezerwacje.NrRezerwacji) AS PoliczOfNrRezerwacji,  
Domki.Garaz FROM Domki INNER JOIN Rezerwacje ON Domki.NrDomku =  
Rezerwacje.NrDomku GROUP BY Domki.Garaz HAVING (((Domki.Garaz)  
Like 'NIE'));
```

e) Odp.:

Nr Domku	Liczba Dni
1	8
2	10
3	7
4	8
5	7
6	5
7	10
8	10
9	7

```
SELECT Rezerwacje.NrDomku, Max(Rezerwacje.LiczbaDni) AS  
MaksimumOfLiczbaDni FROM Rezerwacje GROUP BY  
Rezerwacje.NrDomku;
```

Komentarz:

Zadanie 5 miało charakter bazodanowy. Dane do zadania umieszczono w trzech plikach: domki.txt, pracownicy.txt, rezerwacje.txt. Pliki te odpowiadały trzem tabelom relacyjnej bazy danych. Tabele *domki* i *rezerwacje* oraz *pracownicy* i *rezerwacje* powiązane były relacjami „jeden do wielu” poprzez identyfikator domku oraz identyfikator pracownika. Na zadanie składało się pięć podpunktów a) – e). Realizacja podpunktu a) wymagała uzyskania informacji z jednej tabeli *rezerwacje*, grupowania i sumowania danych. Udzielenie odpowiedzi na podpunkt b) wymagało wydobywania informacji z dwóch tabel z wykorzystaniem połączenia tych tabel oraz filtrowania i sortowania, zaś na podpunkt c) z wszystkich trzech tabel oraz utworzenia formuły (iloczyn ilości dni i ceny za dobę), a następnie wyznaczenia maksymalnej wartości. Realizacja podpunktu d) wymagała skorzystania z dwóch tabel, filtrowania i zastosowania funkcji agregujących. Uzyskanie prawidłowej odpowiedzi na podpunkt e) wymagało grupowania i zastosowania funkcji znajdującej wartość maksymalną w jednej tabeli.

Trudność poszczególnych podpunktów jest podobna, co odzwierciedla punktacja (2 punkty za każdy podpunkt). Zadanie piąte okazało się dla zdających umiarkowanie trudne.

Zadanie 6. Badanie wyników (10 pkt)

W liceum ogólnokształcącym przeprowadzono badanie wyników nauczania z historii. Do tego celu wykorzystano test składający się z 25 pytań, które kolejno dotyczyły poszczególnych epok historycznych:

pytania od 1 do 5	prehistoria
pytania od 6 do 10	starożytność
pytania od 11 do 15	średniowiecze
pytania od 16 do 20	historia nowożytna
pytania od 21 do 25	historia najnowsza

Wyniki testu dla 126 osób umieszczono w pliku `test.txt`. Pierwszy wiersz zawiera nagłówek, składający się z napisu `Nr_ucznia` oraz kolejnych numerów pytań. Kolejne wiersze składają się z numeru ucznia oraz informacji o poprawności jego odpowiedzi na kolejne 25 pytań (0 – niepoprawna odpowiedź lub jej brak, 1 – poprawna odpowiedź).

Dane w wierszach oddzielone są pojedynczymi znakami odstępu.

Przykład:

```
Nr_ucznia 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
1         1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0
2         0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1
3         0 1 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0
```

Korzystając z informacji zawartych w pliku `test.txt` oraz dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych podpunktów umieść w pliku `wyniki6.txt`, poprzedzając je literami oznaczającymi te podpunkty.

- Podaj numery pytań, na które prawidłowo odpowiedziało więcej niż 50% uczniów.
- Podaj liczbę uczniów, którzy w badanej grupie uczniów uzyskali wyniki powyżej średniej liczby poprawnych odpowiedzi.
- Podaj, ilu uczniów otrzymało oceny bardzo dobre, a ilu oceny niedostateczne, przy następującym systemie oceniania:

bardzo dobry	powyżej 90% prawidłowych odpowiedzi
niedostateczny	30% i mniej prawidłowych odpowiedzi

- Podaj numery uczniów, którzy prawidłowo odpowiedzieli na pytania o numerach: 5, 15, 25.
- Utwórz zestawienie, które dla poszczególnych epok historycznych podaje liczbę poprawnych odpowiedzi uczniów na pytania dotyczące danej epoki. Dla utworzonego przez Ciebie zestawienia wykonaj wykres kolumnowy. Pamiętaj o prawidłowym i czytelnym opisie wykresu.

Do oceny oddajesz plik(i) o nazwie(ach),

tu wpisz nazwę(y) pliku(ów)

zawierający(e) komputerową(e) realizację(e) Twoich obliczeń, plik tekstowy `wyniki6.txt` oraz plik o nazwie, zawierający wykres do podpunktu e).

tu wpisz nazwę pliku

Sprawdzane umiejętności

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

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

- posługuje się arkuszem kalkulacyjnym: wykonując obliczenia przy pomocy wbudowanych funkcji i zaprojektowanych formuł,
- 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.	sumuje prawidłowe odpowiedzi i liczy liczbę sum większych od 63.	1	0,47	0,55	–	0,43
6.b.	oblicza średnią i liczy liczbę uczniów, którzy uzyskali wyniki wyższe od średniej.	1	0,45	0,51	–	0,42
6.c.	oblicza zakresy procentowe i liczbę wyników z podanych zakresów.	2	0,60	0,66	–	0,57
6.d.	filtruje dane.	2	0,47	0,55	–	0,42
6.e.	sumuje prawidłowe odpowiedzi w podanych zakresach. Tworzy wykres do zestawu danych.	4	0,42	0,46	–	0,40

Przykładowy poprawny zapis rozwiązania:

a) Odp.: 2, 9, 12,18,19,20,22

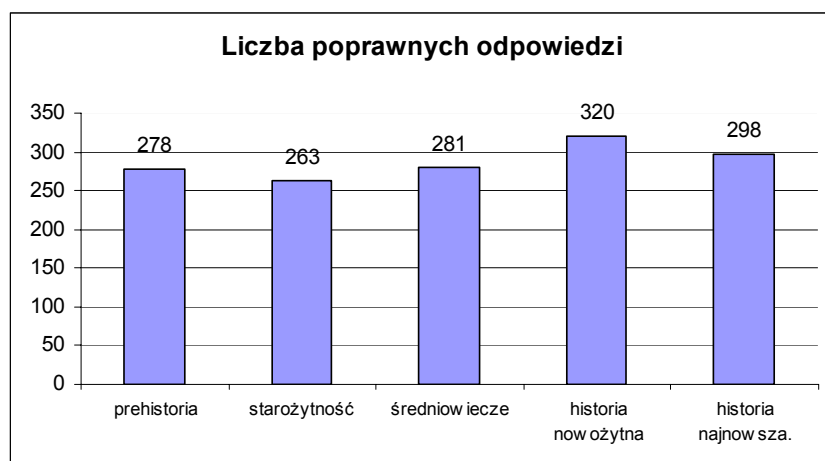
b) Odp.: 55

c) Odp.: bdb – 1, ndst – 6

d) Odp.: 2, 45, 65, 70 i 79

e)

epoka historyczna	liczba poprawnych odpowiedzi
prehistoria	278
starożytność	263
średniowiecze	281
historia nowożytna	320
historia najnowsza	298



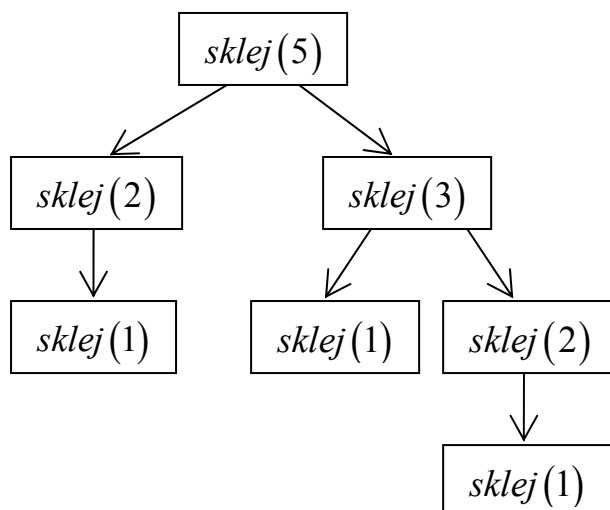
Zadanie 6 składa się z pięciu podpunktów a) – e). Wszystkie dotyczą zestawu danych z tabeli opisującej wynik sprawdzianu z 25 pytaniami (dostarczonej w postaci pliku tekstowego). Dane składają się z wierszy zawierających po 26 liczb (pierwsza liczba to numer ucznia, pozostałe to jedynki i zera, oznaczające poprawne bądź niepoprawne rozwiązanie poszczególnych zadań). Naturalnym narzędziem do rozwiązania tego zadania jest arkusz kalkulacyjny. Uzyskanie odpowiedzi dla punktów a) i b) było stosunkowo proste, wymagało jedynie zastosowania standardowych funkcji w odniesieniu do wierszy/kolumn tabeli i funkcji zliczających w oparciu o proste warunki logiczne. Nieco trudniejszy, choć wymagający analogicznych umiejętności jest punkt c). Aby rozwiązać punkt d), wystarczyło wykorzystać filtrowanie. Najbardziej pracochłonny jest punkt e), wymagający pogrupowania i podsumowania kolumn i stworzenia wykresu ilustrującego uzyskane wyniki.

Arkusz I

Opisana poniżej funkcja rekurencyjna wyznacza, dla liczby naturalnej $n > 0$, długość napisu uzyskanego przez sklejenie binarnych reprezentacji liczb naturalnych od 1 do $n-1$.

krok 3. jeśli n nieparzysta, to wynikiem jest $n-1 + \text{sklej}((n-1)/2) + \text{sklej}((n+1)/2)$

a) Wykonanie funkcji *sklej* można przedstawić w postaci drzewa wywołań rekurencyjnych, ilustrującego wszystkie wywołania funkcji po jej uruchomieniu dla zadanego argumentu. Poniższy rysunek przedstawia takie drzewo dla wywołania *sklej*(5).



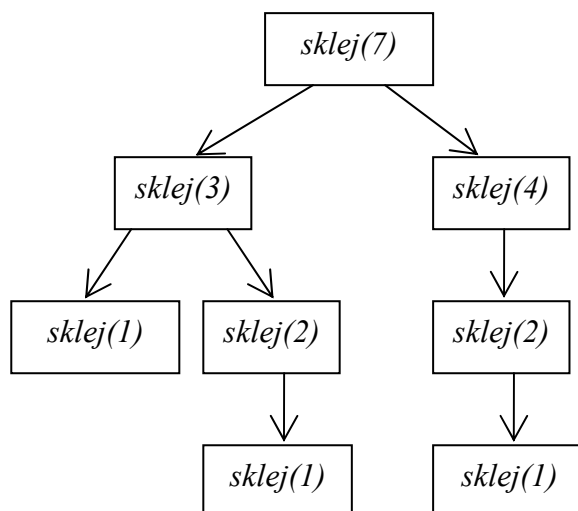
Narysuj analogiczne drzewo dla wywołania `sklej(7)`.

[illegible]

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.a	rysuje drzewo wywołań rekurencyjnych dla opisanej funkcji.	1	0,77	0,82	–	0,57
1.b	podaje wartości funkcji dla wskazanych argumentów.	2	0,82	0,86	–	0,67
1.c	tworzy własny algorytm.	4	0,70	0,77	–	0,43

Przykładowy poprawny zapis rozwiązania:

1.a)



1.b)

n	$sklej(n)$
1	0
2	1
3	3
4	5
5	8
6	11

1.c)

rozwiązanie 1

```

1. s[1] ← 0
2. i ← 2
3. dopóki i ≤ n
    3.1. jeśli (i mod 2) = 0 to s[i] ← i-1+2*s[i/2]
        w przeciwnym wypadku s[i] ← i-1+s[(i-1)/2]+s[(i+1)/2]
    3.2. i←i+1
4. wypisz tablicę s[1..n]
  
```

rozwiązanie 2

```

int dodaj=1;
  
```

```

s[1]=0;
for (int a=2; a<=n; a++)
{
    s[a]=s[a-1]+dodaj;
    if ((a&(a-1))==0)
        dodaj++;
}

```

Komentarz:

Zadanie pierwsze podzielone zostało na trzy 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 przedstawić drzewo wywołań rekurencyjnych dla zadanego parametru wywołania, natomiast w drugim należało wyznaczyć wartości funkcji dla pierwszych sześciu liczb naturalnych. Ostatni, trudniejszy podpunkt wymagał samodzielnego skonstruowania i zapisu algorytmu. Dla ułatwienia w zadaniu podano wskazówki naprowadzające maturzystę na rozwiązanie.

Zadanie okazało się dla uczniów stosunkowo łatwe, w najpopularniejszym rozwiązaniu maturzyści korzystali ze wzorów zawartych w podanej funkcji rekurencyjnej (rozwiązanie 1). Pojawiały się również ciekawsze algorytmy (rozwiązanie 2), obliczające długość napisu uzyskanego przez sklejenie binarnych reprezentacji liczb naturalnych.

Zadanie 2. Potęgowanie (5 pkt)

Dana jest następująca specyfikacja oraz algorytm obliczania potęgi o wykładniku naturalnym:

Specyfikacja:

Dane: liczba rzeczywista a oraz liczba naturalna n , $n \neq 0$

Wynik: liczba rzeczywista $p = a^n = \underbrace{a \cdot a \cdot a \cdot \dots \cdot a}_{n \text{ razy}}$

Algorytm:

krok 1. $p := 1$, $b := a$

krok 2. dopóki $n > 0$ wykonuj:

- a) jeśli $n \bmod 2 \neq 0$, to $p := p * b$
- b) $b := b * b$
- c) $n := n \div 2$

Uwaga: $n \div 2$ oznacza wynik dzielenia całkowitego n przez 2, a $n \bmod 2$ oznacza resztę z dzielenia całkowitego n przez 2.

- a) Przeanalizuj podany algorytm i uzupełnij tabelę wartościami zmiennych p , b oraz n po kolejnych wykonaniach kroku 2 dla dowolnej początkowej wartości a oraz dla początkowej wartości zmiennej n równej 12.

p	b	n
1	a	12

1	a^2	

- b) Uzupełnij poniższą tabelę, wpisując liczby wszystkich mnożeń, wykonywanych przez powyższy algorytm dla podanych wartości n , tzn. liczby wykonanych instrukcji $p := p * b$ i $b := b * b$.

n	liczba mnożeń
2	
3	
4	
5	
6	
7	

- c) Podkreśl funkcję, której wartość jest równa liczbie mnożeń wykonywanych przez powyższy algorytm dla wartości n będącej potęgą dwójki:

- $f(n) = 2 + \log_2 n$
- $f(n) = 1 + n$
- $f(n) = 2n^2 - 1$
- $f(n) = 2^n$

Sprawdzane umiejętności

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

- zna techniki algorytmiczne i algorytmy,
- potrafi otrzymać kolejne wartości funkcji dla wskazanych argumentów,
- wyznacza liczbę wykonywanych operacji.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
2.a	analizuje algorytm, podaje wartości zmiennych.	2	0,68	0,75	–	0,39
2.b	podaje liczbę operacji	2	0,79	0,82	–	0,65

	mnożenia.					
2.c	wskazuje zależność pomiędzy potęgą dwójki a liczbą mnożeń wykonywanych przez algorytm.	1	0,88	0,93	–	0,65

Poprawny zapis rozwiązania:

2.a)

p	b	n
1	a	12
1	a^2	6
1	a^4	3
a^4	a^8	1
a^{12}	a^{16}	0

2.b)

n	liczba mnożeń
2	3
3	4
4	4
5	5
6	5
7	6

2.c)

$$(f(n) = 2 + \log_2 n)$$

Komentarz:

W zadaniu drugim podano w postaci iteracyjnej algorytm szybkiego potęgowania. Zdający miał za zadanie przeanalizować jego działania oraz wskazać liczbę wykonywanych operacji mnożenia. Zadanie składało się z trzech punktów a) – c), z których pierwsze dwa pozwalały (słabo obeznanym maturzystom) oswoić się z ideą algorytmu. W pierwszym punkcie zdający analizował wartości zmiennych pojawiające się w kolejnych iteracjach głównej pętli algorytmu dla dwunastej potęgi dowolnej liczby. W punkcie b) należało natomiast wyznaczyć liczbę mnożeń wykonywanych przez algorytm dla wykładników z zakresu od 2 do 7. Na koniec, w punkcie c), zdający powinien uogólnić obserwacje z b), wybierając funkcję określającą liczbę mnożeń w zależności od wartości wykładnika.

Pojawiające się nieprawidłowe odpowiedzi wynikały z błędów w działaniach matematycznych, np. uczniowie, mnożąc dwie potęgi o takich samych podstawach, mnożyli wykładniki potęg zamiast je dodać.

Zadanie 3. Test (8 pkt)

Podpunkty a) – h) zawierają po cztery odpowiedzi, z których każda jest albo prawdziwa, albo fałszywa. Zdecyduj, które z podanych odpowiedzi są prawdziwe (P), a które fałszywe (F). Zaznacz znakiem X odpowiednią rubrykę w tabeli.

a) Liczba 21202_3 jest równa

	P	F
$D1_{16}$		
321_8		
10110001_2		
211_{10}		

b) Rozważ algorytm, który dla zadanego naturalnego $n > 0$ oblicza następującą sumę:

$$suma = 1^1 + 2^2 + 3^3 + 4^4 + \dots + n^n$$

Algorytm:

krok 1. $suma := 1, i := 2$

krok 2. dopóki $i \leq n$, wykonuj

a. $j := i, p := 1$

b. dopóki $j \geq 1$, wykonuj:

(i) $p := p * i$

(ii) $j := j - 1$

(iii) $suma := suma + p, i := i + 1$

Oceń prawdziwość stwierdzeń:

	P	F
Liczba instrukcji wykonana przez ten algorytm nie zależy od wielkości n .		
Liczba instrukcji wykonana przez ten algorytm jest funkcją kwadratową ze względu na n .		
Instrukcja w kroku 2. jest instrukcją iteracji.		
Wartość zmiennej j w kroku 2.b. zmienia się kolejno od 1 do i , przy $n > 1$.		

c) Algorytmy kryptograficzne dotyczą

	P	F
kompresji danych.		
szyfrowania danych.		
zapewnienia bezpieczeństwa przesyłanych informacji.		
przekształcania obrazów.		

d) Strategia przeszukiwania liniowego

	P	F
może być wykorzystana do znalezienia najmniejszego elementu w ciągu liczb.		
może być wykorzystana do sprawdzenia, czy dany znak występuje w tekście.		
jest wykorzystywana do obliczania wartości silni.		
jest najbardziej efektywną metodą znajdowania elementu w uporządkowanym ciągu.		

e) Aby program napisany w języku programowania wysokiego poziomu mógł być wykonany przez komputer,

	P	F
musi być przetłumaczony na język wewnętrzny komputera.		

musi być wydrukowany.		
niezbędny jest dostęp do kompilatora lub interpretera tego języka.		
wystarczy zmienić rozszerzenie głównego pliku tego programu na exe.		

f) Grafika wektorowa jest wykorzystywana do reprezentowania

	P	F
schematów i kompozycji figur geometrycznych.		
czcionek.		
zdjęć wysokiej jakości.		
obrazów pochodzących bezpośrednio ze skanera.		

g) Pamięć operacyjna komputera

	P	F
jest wykorzystywana wyłącznie przez aplikacje służące do administrowania systemem operacyjnym.		
służy głównie jako nośnik do archiwizacji dokumentów.		
jest wykorzystywana do przechowywania programu komputerowego podczas jego uruchamiania i wykonywania.		
jest niezbędna do poprawnej pracy komputera.		

h) Protokół sieciowy

	P	F
SSL umożliwia bezpieczne przesyłanie danych w sieci.		
IP używany jest tylko w sieci lokalnej.		
POP3 to protokół odbioru poczty elektronicznej.		
HTTP dotyczy przesyłania dokumentów zapisanych w języku HTML.		

Sprawdzane umiejętności

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

- zna systemy liczbowe mające zastosowanie w informatyce,
- zna techniki algorytmiczne i algorytmy,
- zna wybrane struktury danych, w tym podstawowe pojęcia związane z językiem programowania, charakteryzuje sposoby reprezentowania informacji w komputerze,
- zna sposoby reprezentowania informacji w komputerze,
 - zna rolę, funkcje i zasady pracy sprzętu komputerowego.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
3.a	zapisuje liczbę w różnych systemach liczbowych.	1	0,65	0,70	–	0,48
3.b.	analizuje algorytm.	1	0,37	0,42	–	0,17
3.c.	zna techniki algorytmiczne i algorytmy.	1	0,89	0,91	–	0,83
3.d.	zna techniki algorytmiczne i algorytmy.	1	0,62	0,71	–	0,26

3.e.	zna wybrane struktury danych, w tym podstawowe pojęcia związane z językiem programowania.	1	0,87	0,88	–	0,83
3.f.	zna specyfikę i wykorzystanie grafiki wektorowej.	1	0,64	0,68	–	0,48
3.g.	zna i opisuje pojęcie pamięci komputerowej.	1	0,96	0,94	–	1,00
3.h.	zna i opisuje pojęcie protokołu sieciowego.	1	0,58	0,63	–	0,39

Poprawny zapis rozwiązania:

a) PPFF, b) FPPF, c) FPPF, d) PPFF, e) PFPF, f) PPFF, g) FFPP, h) PFPF

Komentarz:

Zadanie miało charakter testu wyboru, sprawdzającego znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Składało się 8 pytań a) – h) wielokrotnego wyboru (wybór z czterech odpowiedzi). Pytania a), b) i d) dotyczyły algorytmiki i związanych z nią pojęć, konkretniej oceny złożoności czasowej algorytmu (punkt b), znajomości metod zamiany między pozycyjnymi systemami liczbowymi, znajomości pojęcia przeszukiwania liniowego. Punkty c), e) i f) dotyczyły pojęć związanych z szeroko rozumowanym oprogramowaniem i kodowaniem danych (algorytmy kryptograficzne, kompilatory i grafika wektorowa). Na koniec, w punktach g) i h), sprawdzana była podstawowa wiedza na temat struktury pamięci współczesnych systemów komputerowych i protokołów sieciowych.

Poziom rozszerzony

Arkusz II

Zadanie 4. Trawniki (10 pkt)

Firma „Zielone Miasto” podpisała umowę na utrzymanie trawników dużej aglomeracji od dnia 1 kwietnia do 30 października 2011 roku. Zadaniem firmy jest:

- wywożenie z miasta skoszonej trawy,
- koszenie trawników.

1 kwietnia 2011 roku rano zgromadzone było $10\,000\text{ m}^3$ skoszonej trawy. Firma dysponuje **30 samochodami** do wywozu skoszonej trawy z miasta. Objętość zgromadzonej trawy zmienia się codziennie w następujący sposób:

- przed południem (zaczynając od 1 kwietnia) każdy samochód firmowy wywozi **15 m^3** skoszonej trawy z miasta,
- w ciągu każdego dnia kosi się **600 m^3** trawy,
- w nocy objętość zgromadzonej trawy zmniejsza się o **3%**. Ubytek trawy **zaokrąglamy w dół do całkowitej liczby m^3** . Na przykład, jeśli wieczorem zgromadzono 60 m^3 trawy, to w nocy jej objętość zmniejszy się o 1 m^3 , co odpowiada liczbie $3\% \cdot 60\text{ m}^3$ zaokrąglonej w dół do liczby całkowitej.

Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe polecenia. Odpowiedzi umieść w pliku zadanie4.txt, poprzedzając je literami oznaczającymi odpowiednie podpunkty.

- Podaj, o ile m^3 zmniejszy się objętość zgromadzonej trawy w nocy z 9 na 10 kwietnia 2011 roku.
- Wskaż, poprzez wstawienie znaku X w odpowiednich wierszach, które z poniższych zdań staną się prawdziwe po uzupełnieniu ich odpowiednimi liczbami. Zdania prawdziwe uzupełnij.

	Prawda
1. Po dniach po raz pierwszy objętość zgromadzonej trawy rano była większa niż rano dnia poprzedniego. Jest to dzień (wpisz datę):	
2. Po dniach objętość zgromadzonej trawy rano wyniesie po raz pierwszy tyle samo, co dnia poprzedniego o tej samej porze. Jest to dzień (wpisz datę):	
3. Najmniejsza liczba samochodów potrzebna firmie, żeby 12 kwietnia 2011 roku po raz pierwszy została wywieziona cała zgromadzona rano trawa, wynosi	

- Wyznacz objętości zgromadzonej trawy po 100 dniach pracy firmy (rano, w 101 dniu, przed wywożeniem trawy) przy założeniu, że 1 kwietnia 2011 rano zgromadzone było odpowiednio $10\ 000\ m^3$, $7\ 000\ m^3$, $4\ 000\ m^3$ skoszonej trawy. Sporządź wykres kolumnowy porównujący te wartości. Zadbaj o czytelność wykresu (pamiętaj o tytule i opisie danych).

Do oceny oddajesz plik(i) o nazwie(ach), zawierający(e)
u wpisz nazwę(y) pliku (ów)

komputerową realizację Twoich obliczeń, plik tekstowy zadanie4.txt, zawierający odpowiedzi do podpunktów a) i c) zadania (odpowiedź do każdego podpunktu poprzedź literą

oznaczającą odpowiedni podpunkt) oraz plik o nazwie
tu wpisz nazwę pliku

zawierający wykres do zadania c).

Sprawdzane umiejętności

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

- projektuje i przeprowadza wszystkie etapy na drodze do otrzymania informatycznego rozwiązania problemu,
- analizuje procesy i zjawiska oraz ocenia możliwość ich komputerowego rozwiązania,
- korzysta ze środków informatyki do przygotowania dokumentu obrazującego graficznie informacje adekwatne do jej charakteru.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
4.a.	zapisuje w postaci formuł proces opisany w zadaniu.	2	0,53	0,58	–	0,33

4.b.	odczytuje wartości zasymulowanego procesu dla określonych danych.	3	0,40	0,44	–	0,22
4.c.	symuluje przebieg procesu dla różnych wartości początkowych; tworzy wykres.	5	0,58	0,63	–	0,39

Przykładowy poprawny zapis rozwiązania:

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K
1		10000	wartość początkowa		30	liczba samochodów					
2											
3	DATA	rano	po wywozie	po koszeniu		zmiana rano					
4	2011-04-01	10000	9550	10150							
5	2011-04-02	9846	9396	9996		-154					
6	2011-04-03	9697	9247	9847		-149					
7	2011-04-04	9552	9102	9702		-145					
8	2011-04-05	9411	8961	9561		-141		a.	271		
9	2011-04-06	9275	8825	9425		-136		b.	2011-10-03	185	
10	2011-04-07	9143	8693	9293		-132					
11	2011-04-08	9015	8565	9165		-128					
12	2011-04-09	8891	8441	9041		-124					
13	2011-04-10	8770	8320	8920		-121					
14	2011-04-11	8653	8203	8803		-117					
15	2011-04-12	8539	8089	8689		-114					
16	2011-04-13	8429	7979	8579		-110					

Zapisane formuły i funkcje:

$B5=D4-ZAOKR.DO.CAŁK(D4*3/100)$

$C4=B4-(15* \text{sam})$ gdzie sam to nazwa komórki E1

$D4=C4+600$

Odp. na a)

271

$I8= \$D\$12- \$B\13

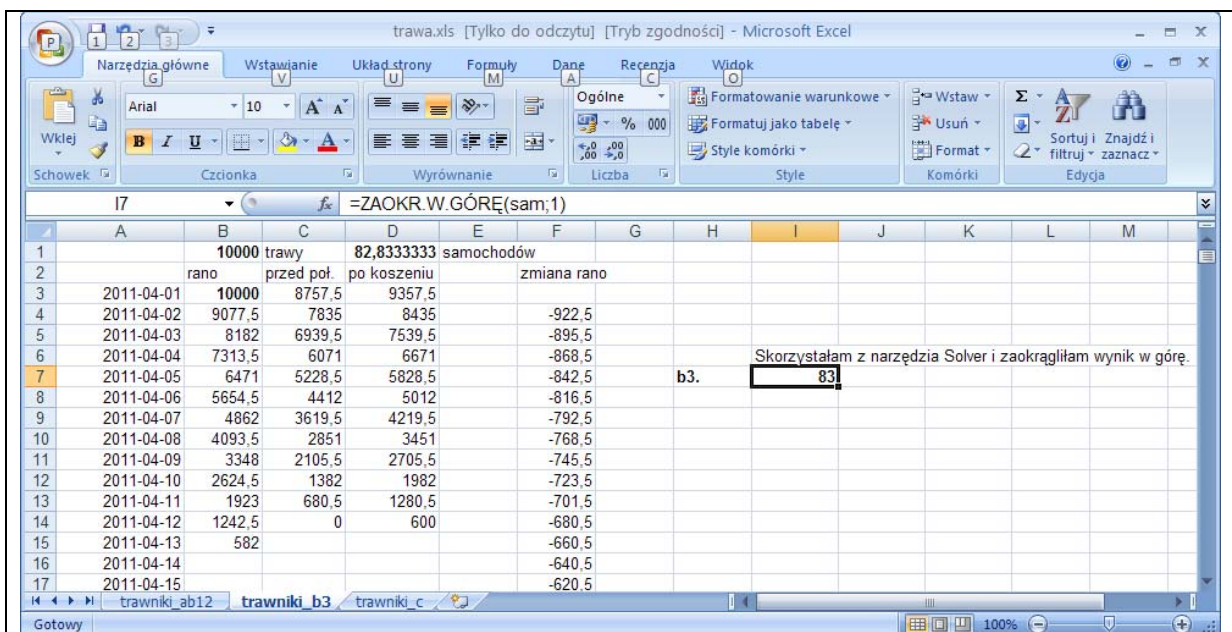
Odp. na b)

Po 185 dniach, 3.10.2011

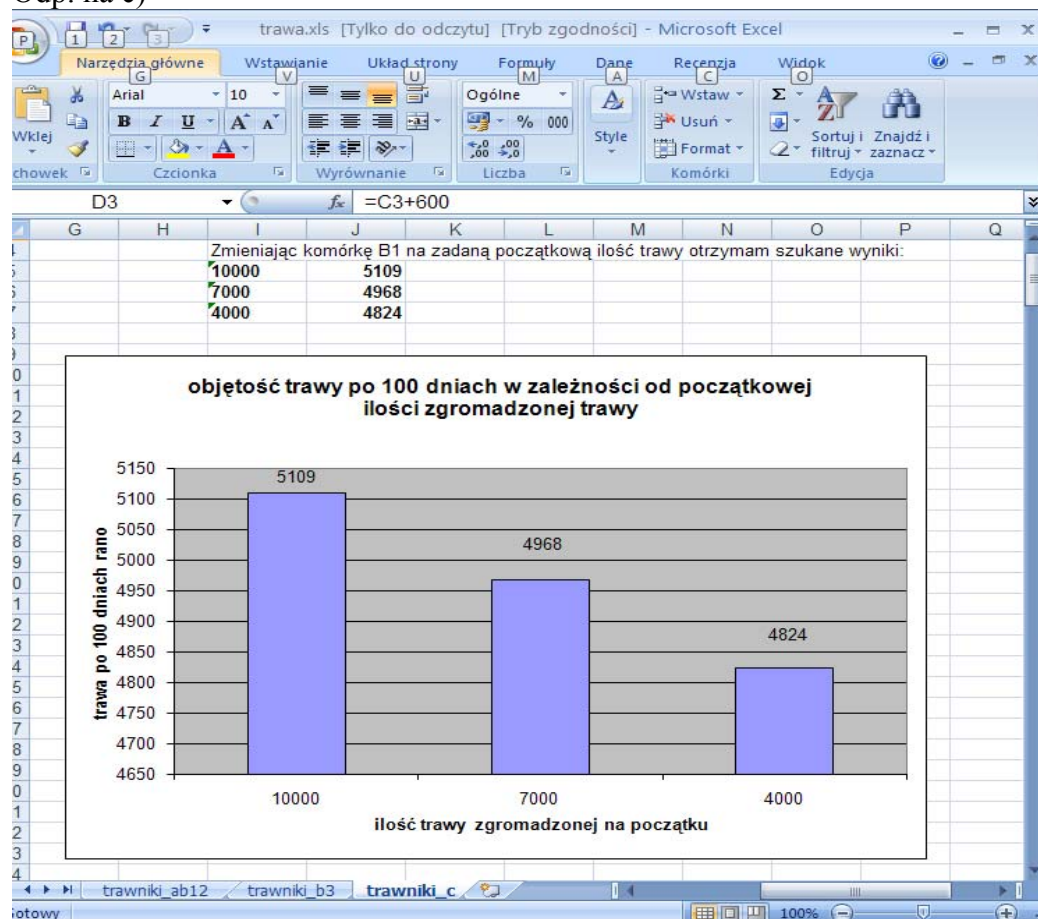
$I9=INDEKS(A:A;PODAJ.POZYCJĘ(0;F:F;0))$

$J9=I9-A4$

Najmniejsza liczba samochodów wynosi 83



Odp. na c)



Komentarz:

Zadanie 4 ma charakter symulacyjny. Opisano w nim fikcyjny proces zmian objętości skoszonej trawy zgromadzonej przez firmę utrzymującą miejskie trawniki. Następnie w trzech punktach a) – c) przedstawiono polecenia dla maturzysty. Kluczowym krokiem w kierunku rozwiązania wszystkich podpunktów było zasymulowanie procesu opisanego w zadaniu, co bez większych trudności można zrobić przy pomocy arkusza kalkulacyjnego. Następnie,

z wykorzystaniem odpowiednich funkcji logicznych i arytmetycznych, można uzyskać informacje wymagane w rozwiązaniach punktów a), b) i c). W ostatniej części, oprócz wyznaczenia odpowiednich wartości liczbowych, należało sporządzić wykres kolumnowy. W zadaniu, pomimo prawidłowego zapisu formuł obliczających wartości w opisanym procesie, wielu maturzystów popełniało błędy obliczeniowe w liczbie dni (różnica 1 dnia) czy liczbie samochodów.

Zadanie 5. Wystawy psów (10 pkt)

Pewien klub kynologiczny przechowuje w plikach tekstowych informacje o swoich członkach i ich psach. Informacje przechowywane w wierszach w każdym z plików są rozdzielone średnikami.

W pliku *osoby.txt* każdy wiersz zawiera informacje o członkach klubu: *id_osoby*, *imię*, *nazwisko*, *nr telefonu*.

Przykład:

```
o001;Adam;Roztoka;501358358
o002;Adam;Komarnicki;337234875
```

Plik *psy.txt* zawiera informacje dotyczące psów i ich właścicieli: *id_psa*, *rasa*, *wiek*, *pleć*, *liczba zdobytych medali*, *id_osoby*.

Przykład:

```
1;wyżeł węgierski krótkowłosey;4;samica;1;o059
2;owczarek niemiecki;2;samica;2;o064
3;chihuahua;3;samiec;4;o097
```

Wykorzystując dane zawarte w tych plikach oraz dostępne narzędzia informatyczne, wykonaj poniższe polecenia. Odpowiedzi zapisz w pliku *zadanie5.txt*, poprzedzając je literami oznaczającymi odpowiednie podpunkty.

- Podaj liczbę samców oraz liczbę samic wśród psów.
- Utwórz zestawienie podające nazwiska i imiona osób, które mają więcej niż 8 psów. Zestawienie powinno być uporządkowane alfabetycznie według nazwisk.
- Podaj imię i nazwisko osoby, której psy zdobyły łącznie najwięcej medali, oraz podaj liczbę tych medali.
- Podaj liczbę osób posiadających owczarki. Zwróć uwagę na to, że nazwa rasy może składać się z kilku wyrazów oraz że jedna osoba może posiadać kilka owczarków tej samej rasy lub różnych ras.

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 *zadanie5.txt*, zawierający odpowiedzi do podpunktów a), b), c), d) zadania (odpowiedź do każdego podpunktu poprzedź literą oznaczającą podpunkt).

Sprawdzane umiejętności

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

- projektuje relacyjne bazy danych i proste aplikacje bazodanowe,
- stosuje metody wyszukiwania i przetwarzania informacji w relacyjnych bazach danych,
- tworzy proste aplikacje bazodanowe, wykorzystuje język zapytań,

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
5.a.	filtruje w jednej tabeli.	2	0,91	0,93	–	0,83
5.b.	wyszukuje informacje z dwóch tabel, grupuje i	3	0,78	0,83	–	0,58

	sortuje.					
5.c.	wyszukuje informacje z dwóch tabel, grupuje i sortuje.	2	0,68	0,71	–	0,54
5.d.	filtruje z zastosowaniem znaków globalnych, grupuje.	3	0,35	0,39	–	0,17

Przykładowy poprawny zapis rozwiązania:

a) Odp: 492 samce, 508 samic

```
SELECT Psy.płeć, Count(Psy.idPsa) AS PoliczOfidPsa FROM Psy
GROUP BY Psy.płeć;
```

b)

nazwisko	imię
Antczak	Edyta
Bielawska	Dagmara
Bugajno	Kamila
Gabowski	Pawel
Jaworski	Eugeniusz
Kaliszewski	Kacper
Kowal	Marcin
Kowalski	Ryszard
Kwiatkowski	Marcin
Szargawinska	Danuta
Szczupak	Anna
Tobera	Krystyna
Wroclawski	Rafal
Wysocka	Maria

```
SELECT Osoby.Nazwisko, Osoby.Imię FROM Osoby INNER JOIN Psy ON
Osoby.idOsoby = Psy.idOsoby1 GROUP BY Osoby.Nazwisko,
Osoby.Imię HAVING (((Count(Psy.idPsa))>8)) ORDER BY
Osoby.Nazwisko;
```

c)

Odp: Krystyna Tobera, 41 medali

```
SELECT TOP 1 Osoby.Imię, Osoby.Nazwisko, Sum(Psy.medali) AS
SumaOfmedali FROM Osoby INNER JOIN Psy ON Osoby.idOsoby =
Psy.idOsoby1 GROUP BY Osoby.Imię, Osoby.Nazwisko ORDER BY
Sum(Psy.medali) DESC;
```

d)

Odp: 103 osoby

Kwerenda pomocnicza:

```
SELECT Osoby.idOsoby, Count(Psy.rasa) AS PoliczOfrasa FROM
Osoby INNER JOIN Psy ON Osoby.idOsoby=Psy.idOsoby1 WHERE
(((Psy.rasa) Like "*owczarek*")) GROUP BY Osoby.idOsoby;
```

Kwerenda z wynikiem:

```
SELECT Count(Kwerenda_pom4.idOsoby) AS [osoby posiadające
owczarka(i)] FROM Kwerenda_pom4;
```

Komentarz:

Zadanie miało charakter bazodanowy. Dane do zadania umieszczono w dwóch plikach opisujących psy i ich właścicieli. Pliki te odpowiadały tabelom relacyjnej bazy danych, powiązanych poprzez identyfikator właściciela psa (IDosoby). 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 samcami, a następnie z samicami. W podpunkcie b) należało policzyć, ile psów ma każdy właściciel, a następnie odfiltrować tych, którzy mieli więcej niż 8 psów i posortować właścicieli według nazwisk. Podpunkty c) i d), okazały się trudniejsze dla zdających. Rozwiązanie wymagało łączenia tabel, sortowania, filtrowania i grupowania danych. W podpunkcie c) zdający nieprawidłowo obliczali liczbę medali, zaś w podpunkcie d) popełniali błędy w odfiltrowaniu osób posiadających owczarki, nieprawidłowo stosując znak „*”. Zdarzało się, choć dużo rzadziej niż w latach ubiegłych, że uczniowie zamieszczali tylko plik tekstowy z odpowiedziami, nie dołączając plików realizacji obliczeń.

Zadanie 6. Liczby inaczej (10 pkt)

W pliku `liczby.txt`, w oddzielnych wierszach, znajduje się 1000 liczb zapisanych w systemie dwójkowym o długościach zapisów od 2 do 16 cyfr (0 lub 1).

Napisz program, którego wykonanie da odpowiedzi do poniższych podpunktów. Odpowiedzi zapisz w pliku `zadanie6.txt`, a każdą odpowiedź poprzedź literą oznaczającą ten podpunkt.

- Ile jest liczb parzystych w całym pliku?
- Jaka jest największa liczba w tym pliku? Podaj jej wartość w dwóch systemach: dwójkowym i dziesiętnym.
- Ile liczb w całym pliku ma dokładnie 9 cyfr? Podaj sumę tych liczb w systemie dwójkowym.

Do oceny oddajesz plik `zadanie6.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 III obszaru standardów. Zdający:

- stosuje kolejne etapy prowadzące do otrzymania poprawnego rozwiązania: od sformułowania specyfikacji problemu po testowanie rozwiązania,
- wykorzystuje metody informatyki w rozwiązywaniu problemów.

Lp.	Oceniane czynności Zdający:	Liczba pkt	Łatwość czynności			
			Ogółem	LO	LP	T
1.	oblicza liczbę parzystych liczb w podanym pliku.	2	0,53	0,57	–	0,39
2.	konwertuje liczbę binarną na dziesiętną, wyszukuje maksymalną liczbę w pliku.	4	0,42	0,44	–	0,35
3.	wyszukuje liczby binarne 9-cyfrowe oraz sumuje te liczby.	4	0,41	0,44	–	0,28

Przykładowy poprawny zapis rozwiązania:

a)

Odp.: liczba parzystych liczb – 497

b)

Odp.: maksymalna liczba w systemie dziesiętnym – 64502

maksymalna liczba w systemie dwójkowym – 1111101111110110

c)

Odp.: liczba liczb o długości 9 – 101

suma tych liczb w systemie dwójkowym – 1001100111110111

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

using namespace std;

string wynik[20]; // 6c. suma liczb w systemie dwójkowym

void dec2bin(float a) { // zamienia liczbe dziesiętna na
    dwójkowa i zapisuje ja w tablicy wynik
    float b = 2; // baza systemu dwójkowego
    for (int i=0;i<=16;i++) {
        if (a >= pow(b,16-i)) {
            wynik[i] = '1';
            a -= pow(b,16-i);
        }
        else {
            wynik[i]= '0';
        }
    }
}

int bin2dec(string s) { // zamienia liczbe dwójkowa o typie
    string na dziesiętna o typie int
    int t = 0; // zmienna pomocnicza
    int l; // liczba cyfr w zapisie dwójkowym
    float a = 2; // baza systemu

    l = s.length();
    for (int i=1;i<=l;i++) {
        if (s[l-i]=='1') t += pow(a,i-1);
    }
    return t;
}

int main() {
    ifstream in ("liczby.txt");
    ofstream ou ("zadanie6.txt");
    int a;
    int la = 0; // licznik liczb parzystych.
    int lb = 0; // największa liczba.
    string bb; // największa liczba binarnie
    int lc = 0; // licznik liczb 9-cyfrowych
    int sc = 0; // suma cyfr 9-cyfrowych
    int l; // dlugosc linii
    string s;
    char t = '0';
    int i=1;
```

```

while (in >> s) {
    l = s.length();
    a = bin2dec(s);
    if ((a%2)==0) la++;
    if (a>lb) {
        lb = a;
        bb = s;
    }
    if (l==9) {
        lc++;
        sc += a;
    }
}

dec2bin(sc);

ou << "a. \n" << la << endl;
ou << "b. \n" << bb << endl << lb << endl;
ou << "c. \n" << lc << endl;
for (int i=0;i<=16;i++) ou << wynik[i];
ou << endl;

return 0;
}

```

Komentarz:

Zadanie 6 miało charakter programistyczny. Wyróżniało się tym, iż w treści wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego. Algorytmicznie proste zadanie należało zaimplementować w wybranym języku programowania. Zdający otrzymał plik tekstowy składający się z 1000 liczb zapisanych w systemie binarnym. W podpunktach a) – c) tego zadania wymagane było zliczanie liczb parzystych, porównywanie reprezentacji binarnych i konwersja z systemu binarnego do dziesiętnego (punkty a i b), sumowanie liczb zapisanych binarnie lub konwersja z systemu dziesiętnego do binarnego (punkt c). Wskaźniki łatwości dla tego zadania klasyfikują je jako zadanie umiarkowanie trudne.

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. 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 zadaniami, w innych wyniki były znacząco niższe.

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

W dużej mierze wyniki osiągnięte 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 oraz wymagania egzaminacyjne),
- komentarz do zadań egzaminacyjnych z informatyki wydawany przez OKE lub CKE (w publikacji z przedmiotami matematyczno–przyrodniczymi). Komentarz ten zawiera przykładowe rozwiązania zadań, 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 (plik tekstowy z odpowiedziami oraz dołączone pliki komputerowej realizacji rozwiązania). Do tej pory 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.

W pierwszej połowie lipca na stronie Centralnej Komisji Egzaminacyjnej ukazał 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.