

EGZAMIN MATURALNY OD ROKU SZKOLNEGO 2014/2015

INFORMATYKA POZIOM ROZSZERZONY

**ROZWIĄZANIA ZADAŃ I SCHEMATY PUNKTOWANIA
(A1, A2, A3, A4, A6, A7)**

GRUDZIEŃ 2013

Arkusz I

Zadanie 1. Test (0–5)

Wymagania ogólne	Wymagania szczegółowe
I. [...] wykorzystanie sieci komputerowej [...]. II. Wyszukiwanie [...] informacji z różnych źródeł; opracowywanie za pomocą komputera: rysunków, danych liczbowych [...] III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, z zastosowaniem podejścia algorytmicznego V. Ocena [...] ograniczeń [...] informatyki	1. [...] korzystanie z sieci komputerowej. Zdający: 5) prawidłowo posługuje się terminologią sieciową 2. Wyszukiwanie [...] informacji z różnych źródeł, współtworzenie zasobów w sieci. Zdający: 2) [...] stosuje metody wyszukiwania i przetwarzania informacji w [...] bazie danych (język SQL) 4. Opracowanie informacji za pomocą komputera, w tym rysunków [...] Zdający: 2) określa własności grafiki rastrowej 5. Rozwiązywanie problemów [...], stosowanie podejścia algorytmicznego. Zdający 1) analizuje [...] sytuacje problemowe [...] 6) ocenia własności rozwiązania algorytmicznego [...] 11) [...] stosuje a) algorytmy na liczbach całkowitych 7. [...] ocenia zagrożenia i ograniczenia [...] zastosowań informatyki. Zdający: 3) stosuje normy etyczne i prawne związane z [...] ochroną danych w Internecie

a) (0–1)

Poprawna odpowiedź:

F, F, P, F

Schemat punktowania:

1 pkt – poprawna ocena

0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–1)

Poprawna odpowiedź:

F, P, P, F

Schemat punktowania:

1 pkt – poprawna ocena

0 pkt – inna odpowiedź lub brak odpowiedzi

c) (0–1)

Poprawna odpowiedź:

F, P, P, P

Schemat punktowania:

1 pkt – poprawna ocena

0 pkt – inna odpowiedź lub brak odpowiedzi

d) (0–1)

Poprawna odpowiedź:

P, F, F, P

Schemat punktowania:

1 pkt – poprawna ocena

0 pkt – inna odpowiedź lub brak odpowiedzi

e) (0–1)

Poprawna odpowiedź:

F, P, F, F

Schemat punktowania:

1 pkt – poprawna ocena

0 pkt – inna odpowiedź lub brak odpowiedzi

Zadanie 2. Całkowity pierwiastek kwadratowy (0–6)

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji [...], z zastosowaniem podejścia algorytmicznego	5. Rozwiązywanie problemów i [...] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe 2) stosuje podejście algorytmiczne do rozwiązania problemu 6) ocenia własność rozwiązania algorytmicznego [...], np. zgodność ze specyfikacją [...] 11. Opisuje podstawowe algorytmy i stosuje:[...] c) algorytmy numeryczne, np.: - obliczanie wartości pierwiastka kwadratowego 16) opisuje własności algorytmów na podstawie ich analizy 18) oblicza liczbę operacji wykonywanych przez algorytm

a) (0–2)

Poprawna odpowiedź:

uzupełnienie luki ($k:=k+1$)

uzupełnienie tabeli (5, 32)

Schemat punktowania:

2 pkt – poprawne uzupełnienie instrukcji i tabeli

1 pkt – poprawne uzupełnienie algorytmu lub tabeli

0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–2)

Poprawna odpowiedź:

uzupełnienie luki ($k:=s$)

Schemat punktowania:

- 2 pkt – poprawne uzupełnienie instrukcji
 0 pkt – inna odpowiedź lub brak odpowiedzi

c) (0–2)**Poprawna odpowiedź:**

uzupełnienie tabeli (6 dla $n=32$ oraz 6 dla $n=1024$)

Schemat punktowania:

- 2 pkt – poprawne uzupełnienie tabeli dla $n=32$ oraz dla $n=1024$
 1 pkt – poprawne uzupełnienie tabeli dla $n=32$ lub dla $n=1024$
 0 pkt – inna odpowiedź lub brak odpowiedzi

Zadanie 3. Progi i schody (0–9)

Wymagania ogólne	Wymagania szczegółowe
III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego	5. Rozwiązywanie problemów i [...] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe 2) stosuje podejście algorytmiczne do rozwiązania problemu 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji 6) ocenia własność rozwiązania algorytmicznego [...], np. efektywność działania [...] 11) Opisuje podstawowe algorytmy i stosuje:[...] a) algorytmy na liczbach całkowitych 16) opisuje własności algorytmów na podstawie ich analizy 18) oblicza liczbę operacji wykonywanych przez algorytm

a) (0–1)**Poprawna odpowiedź:**

Wartości w kolejności:

schody do dołu, ich długość, liczba progów

2, 2, 2	3	0
3, 1, 1	3	1
3, 3, 1	3	1
11, 7, 7, 6	4	2
7, 7	2	0
9, 9, 7	3	1

Schemat punktowania:

- 1 pkt – poprawne podanie wszystkich wartości
 0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–2)

Przykład poprawnej odpowiedzi:

```
liczba_progów:=0;  
for i:=1 to n-1 do  
    if a[i] > a[i+1] then liczba_progów := liczba_progów + 1;
```

Schemat punktowania:

- 2 pkt – poprawne napisanie algorytmu
- 1 pkt – napisanie algorytmu nieuwzględniającego przypadków brzegowych
- 0 pkt – błędny algorytm lub brak odpowiedzi

c) (0–4)

Przykład poprawnej odpowiedzi:

```
najw_liczba_progów := 0;  
liczba_progów_w_schodach := 0;  
for i := 1 to n-1 do  
    if a[i] < a[i+1] then  
        { * a[i+1] może być początkiem nowych schodów *}  
        begin  
            if liczba_progów_w_schodach > najw_liczba_progów then  
                najw_liczba_progów := liczba_progów_w_schodach;  
            liczba_progów_w_schodach := 0  
        end  
    else  
        if a[i] > a[i+1] then  
            liczba_progów_w_schodach := liczba_progów_w_schodach+1;  
        end  
    end  
end  
if liczba_progów_w_schodach > najw_liczba_progów then  
    najw_liczba_progów := liczba_progów_w_schodach;
```

Schemat punktowania:

- 4 pkt – poprawne napisanie algorytmu
- 2 pkt – napisanie algorytmu, w którym nie uwzględniono progów w schodach do dołu kończących się ostatnim elementem ciągu
- 0 pkt – błędny algorytm lub brak odpowiedzi

d) (0–2)

Poprawna odpowiedź:

W pesymistycznym przypadku w każdym obrocie pętli wykonywane są dwa porównania: pierwsze służy do wykrycia nowych schodów, a drugie do identyfikacji progów. Zatem w pesymistycznym przypadku algorytm wykona dokładnie $2(n-1)$ porównań.

Schemat punktowania:

- 2 pkt – poprawne podanie liczby porównań i poprawne uzasadnienie
- 1 pkt – poprawne podanie liczby porównań i błędne uzasadnienie lub brak uzasadnienia
- 0 pkt – inna odpowiedź lub brak odpowiedzi

Arkusz II

Zadanie 4. Anagramy cyfrowe (0–10)

Wymagania ogólne	Wymagania szczegółowe
II. [...] opracowywanie za pomocą komputera danych liczbowych III. Rozwiązywanie problemów i podejmowanie decyzji [...] z zastosowaniem podejścia algorytmicznego	5. Rozwiązywanie problemów i [...] stosowanie podejścia algorytmicznego. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe 2) stosuje podejście algorytmiczne do rozwiązania problemu 4) dobiera efektywny algorytm do rozwiązania sytuacji problemowej i zapisuje go w wybranej notacji 5) posługuje się podstawowymi technikami algorytmicznymi 6) ocenia własność rozwiązania algorytmicznego [...], np. efektywność działania [...] 11) Opisuje podstawowe algorytmy i stosuje:[...] b) algorytmy wyszukiwania i porządkowania (sortowania) 12) projektuje rozwiązanie problemu (realizacja algorytmu) i dobiera odpowiednią strukturę danych 21) przeprowadza komputerową realizację algorytmu i rozwiązania problemu 22) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu i uruchamianiu programów 23) stosuje podstawowe konstrukcje programistyczne w wybranym języku programowania[...]

a) (0–4)

Poprawna odpowiedź:

Liczba wierszy z anagramami: 258

Schemat punktowania:

- 4 pkt – poprawne sortowanie cyfr w liczbach i poprawnie obliczona liczba par anagramów
- 2 pkt – poprawne sortowanie cyfr w liczbach i błędnie obliczona liczba par anagramów
- 0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–6)

Poprawna odpowiedź:

Maksymalna liczba anagramów: 11

Schemat punktowania:

- 6 pkt – poprawne sortowanie cyfr w liczbach, zliczenie danych i obliczanie maksimum
- 4 pkt – poprawne sortowanie cyfr w liczbach, zliczenie danych i błędne obliczanie maksimum

- 2 pkt – poprawne sortowanie cyfr w liczbach, błędnie zliczenie danych i obliczenie maksimum
0 pkt – inna odpowiedź lub brak odpowiedzi

Przykładowa realizacja algorytmu w języku C++:

```
#include <algorithm>
#include <fstream>
#include <iostream>
#include <map>

/**
 * zamiana dodatniej liczby całkowitej k na największy jej anagram cyfrowy
 */

int największy_anagram(int k) {
    int cyfry[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    while (k != 0)
    {
        int cyfra = k % 10;
        cyfry[cyfra]++;
        k = k / 10;
    }
    int wynik = 0;
    for (int i = 9; i >= 0; i--)
        for (int j = 0; j < cyfry[i]; j++)
            wynik = wynik * 10 + i;
    return wynik;
}

std::map<int, int> anagramy;
int wiersze_z_anagramami;
int maksimum;

int main() {
    int u, v;
    std::ifstream liczba("dane_anagramy.txt");
    while (liczba >> u >> v)
    {
        int anagram_u = największy_anagram(u);
        int anagram_v = największy_anagram(v);
        if (anagram_u == anagram_v)
            wiersze_z_anagramami++;
        anagramy[anagram_u]++;
        anagramy[anagram_v]++;
        maksimum = std::max(maksimum, anagramy[anagram_u]);
        maksimum = std::max(maksimum, anagramy[anagram_v]);
    }
    std::cout << "wiersze z anagramami: " << wiersze_z_anagramami <<
        std::endl;
    std::cout << "maksimum anagramow: " << maksimum << std::endl;
    return 0;
}
```

Przykładowa realizacja algorytmu w języku Pascal:

```
program AnagramyCyfrowe;
{**
 * zamiana dodatniej liczby całkowitej k na największy jej anagram cyfrowy
 *}
function NajwiekszyAnagram(k: longint): longint;
var
  cyfry: array[0..9] of integer;
  cyfra, i, j: integer;
  wynik: longint;
begin
  for i := 0 to 9 do cyfry[i] := 0;
  while k <> 0 do
    begin
      cyfra := k mod 10;
      cyfry[cyfra] := cyfry[cyfra] + 1;
      k := k div 10
    end;
  wynik := 0;
  for i := 9 downto 0 do
    for j := 1 to cyfry[i] do
      wynik := wynik * 10 + i;
    end;
  NajwiekszyAnagram := wynik
end;

{**
 * struktura danych umożliwiajaca obliczenie maksymalnej liczby liczb, ktore
 * wzajemnie sa swoimi anagramami
 * anagramy[1..2001] - tablica zawierajaca dotychczas odkryte, rozne anagramy
 * liczba_anagramow - liczba dotychczas odkrytych roznych anagramow
 * liczba_wystapien[1..2000] - liczba_wystapien[i] = liczbie dotychczasowych
 * wystapien anagramu anagramy[i]
 *}
var
  anagramy: array [1..2001] of longint;
  liczba_wystapien: array [1..2000] of integer;
  liczba_anagramow: integer;

procedure Ini;
var
  i: integer;
begin
  for i := 1 to 2000 do
    liczba_wystapien[i] := 0;
  liczba_anagramow := 0;
end;

function Ile(a: longint): integer;
{**
 * ile razy dotychczas pojawil sie anagram a
 *}
var
```

```

    i: integer;
begin
    anagramy[liczba_anagramow + 1] := a; { * wartownik *}
    i := 1;
    while anagramy[i] <> a do
        i := i + 1;
    liczba_wystapien[i] := liczba_wystapien[i] + 1;
    if i = liczba_anagramow + 1 then
        liczba_anagramow := i;
        Ile := liczba_wystapien[i]
    end;

    (**
    * program glowny
    *)
var
    u, v: longint;
    maksimum, wystapienia, wiersze_z_anagramami: integer;
    dane_anagramy: text;
begin
    maksimum := 0;
    wiersze_z_anagramami := 0;
    Ini;
    assign(dane_anagramy, 'dane_anagramy.txt');
    reset(dane_anagramy);
    while not eof(dane_anagramy) do
        begin
            readln(dane_anagramy, u, v);
            u := NajwiekszyAnagram(u);
            v := NajwiekszyAnagram(v);
            if u = v then
                wiersze_z_anagramami := wiersze_z_anagramami + 1;
            wystapienia := Ile(u);
            if wystapienia > maksimum then maksimum := wystapienia;
            wystapienia := Ile(v);
            if wystapienia > maksimum then maksimum := wystapienia
        end;
    close(dane_anagramy);
    writeln('wiersze z anagramami: ', wiersze_z_anagramami);
    writeln('maksimum anagramow: ', maksimum);
    readln;
end.

```

Zadanie 5. Rowery (10 punktów)

Wymagania ogólne	Wymagania szczegółowe
II. [...] przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera, danych liczbowych [...] III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, [...]	4. Opracowywanie informacji za pomocą komputera w tym [...] danych liczbowych. Zdający: 4) wykorzystuje arkusz kalkulacyjny [...] 5. Rozwiązywanie problemów [...]. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin 25) dobiera właściwy program użytkowy [...] do rozwiązania zadania

a) (0–2)**Poprawna odpowiedź:**

Typ	Liczba
play-bike	14
rowerki_i_pojazdy	115
Miejski	200
szosowy_miejski	231
szosowy_wyczynowy	259
gorski_sportowy	289
Składany	290
Uniwersalny	291
gorski_rekreacyjny	311

Schemat punktowania:

- 2 pkt – poprawne zestawienie liczby rowerów poszczególnych typów i posortowanie ze względu na liczby rowerów
- 1 pkt – poprawne zestawienie liczby rowerów poszczególnych typów bez sortowania
- 0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–4)**Poprawna odpowiedź:**

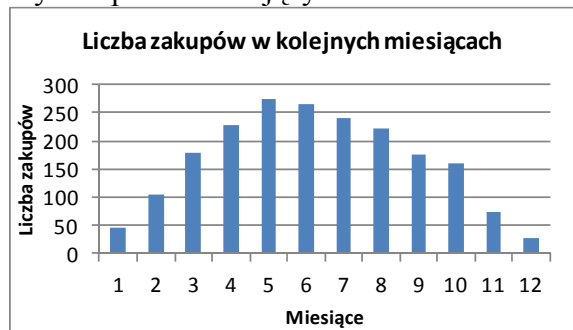
Zestawienie zawierające informacje o liczbie sprzedanych rowerów

Miesiac	Liczba zakupów
1	45
2	105
3	180
4	227
5	274
6	266
7	241
8	223
9	175
10	161
11	74
12	29

Schemat punktowania:

- 2 pkt – poprawne zestawienie liczby sprzedanych rowerów w każdym miesiącu
- 1 pkt – zestawienie z co najmniej sześcioma poprawnymi wierszami
- 0 pkt – inna odpowiedź lub brak odpowiedzi

Wykres przedstawiający zestawienie

**Schemat punktowania:**

- 2 pkt – poprawne wykonanie wykresu (poprawny dobór danych i typu wykresu – 1 pkt, czytelny opis wykresu – 1 pkt)
- 0 pkt – inna odpowiedź lub brak odpowiedzi

Zadanie 6. Serwis samochodowy (0–10)

Wymagania ogólne	Wymagania szczegółowe
II. [...] przetwarzanie informacji z różnych źródeł; opracowywanie za pomocą komputera, danych liczbowych [...] III. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, [...]	2. [...] selekcjonowanie, przetwarzanie [...] informacji, korzystanie z różnych źródeł[...] informacji. Zdający: 1) projektuje relacyjną bazę danych [...] 2) stosuje metody wyszukiwania i przetwarzania informacji w relacyjnej bazie danych (język SQL) 5. Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera [...]. Zdający: 1) analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin 25) dobiera właściwy program użytkowy [...] do rozwiązania zadania

a) (0–2)**Poprawna odpowiedź:**

data
2012-04-13
2012-11-06

Schemat punktowania:

- 2 pkt – podanie poprawnej daty wiosennej i jesiennej wymiany opon
- 1 pkt – podanie jednej poprawnej daty wymiany opon
- 0 pkt – inna odpowiedź lub brak odpowiedzi

b) (0–2)**Poprawna odpowiedź:**

nazwa	PoliczOfnr_rejestr
METALEX	30
FASAT	29
PETROLEUM	25
KONSBET	23
RENOWATOR	19
WERTIX	18
LUBEX	18
FERMILAB	18
FAMUR	17

Schemat punktowania:

- 2 pkt – podanie pełnego zestawienia i poprawne posortowanie
- 1 pkt – podanie pełnego zestawienia bez posortowania
- 0 pkt – inna odpowiedź lub brak odpowiedzi

c) (0–2)**Poprawna odpowiedź:**

miesiac	SumaOfcena
1	15
2	100
3	476
4	1065
5	380
6	195
7	411
8	40
9	40
10	460
11	670
12	70

Schemat punktowania:

- 2 pkt – poprawne podanie pełnego zestawienia
- 0 pkt – inna odpowiedź lub brak odpowiedzi

d) (0–2)**Poprawna odpowiedź:**

numer rejestracyjny: PO 60910

marka samochodu i nazwa firmy: Ford Transit, FERMILAB

Schemat punktowania:

- 2 pkt – poprawne podanie numeru rejestracyjnego i właściciela samochodu
- 1 pkt – poprawne podanie numeru rejestracyjnego lub właściciela samochodu
- 0 pkt – inna odpowiedź lub brak odpowiedzi

e) (0–2)

Poprawna odpowiedź:

nazwa
FAMUR
KONSBET
LUBEX
RENOWATOR

Schemat punktowania:

- 2 pkt – poprawne podanie nazw firm
- 1 pkt – poprawne podanie nazw firm z powtórzeniami
- 0 pkt – inna odpowiedź lub brak odpowiedzi

Przykładowa realizacja w PostgreSQL:

```
CREATE TABLE firma (id VARCHAR NOT NULL PRIMARY KEY,
    nazwa VARCHAR NOT NULL);
CREATE TABLE uslug a (id INTEGER NOT NULL PRIMARY KEY,
    nazwa VARCHAR NOT NULL, cena INTEGER NOT NULL);
CREATE TABLE pojazd (nr_rejestr VARCHAR NOT NULL PRIMARY KEY,
    marka VARCHAR NOT NULL, rok_prod INTEGER NOT NULL,
    firma_id VARCHAR NOT NULL REFERENCES firma);
CREATE TABLE naprawa (id INTEGER NOT NULL PRIMARY KEY,
    data DATE NOT NULL, nr_rejestr VARCHAR NOT NULL REFERENCES pojazd,
    uslug a_id INTEGER NOT NULL REFERENCES uslug a);
```

-- część zależna od platformy

```
¥copy firma FROM firma.txt delimiter E'¥t' csv header;
¥copy uslug a FROM uslug a.txt delimiter E'¥t' csv header;
¥copy pojazd FROM pojazd.txt delimiter E'¥t' csv header;
¥copy naprawa FROM naprawa.txt delimiter E'¥t' csv header;
-- koniec części zależnej od platformy
```

```
SELECT data FROM naprawa JOIN uslug a ON uslug a_id = uslug a.id
WHERE nazwa = 'wymiana opon' AND nr_rejestr = 'PO 3631H'
ORDER BY data;
```

```
SELECT nazwa, COUNT(nr_rejestr)
FROM firma LEFT JOIN pojazd ON firma.id = firma_id
GROUP BY nazwa ORDER BY COUNT(nr_rejestr) DESC, nazwa;
```

```
SELECT EXTRACT(MONTH FROM data) miesiac, SUM(cena) suma
FROM naprawa JOIN uslug a ON uslug a_id = uslug a.id
JOIN pojazd ON naprawa.nr_rejestr = pojazd.nr_rejestr
JOIN firma ON firma_id = firma.id
WHERE firma.nazwa = 'LUBEX'
GROUP BY EXTRACT(MONTH FROM data)
ORDER BY EXTRACT(MONTH FROM data);
```

```
SELECT nr_rejestr, marka, nazwa
FROM pojazd JOIN firma ON firma_id = firma.id
WHERE nr_rejestr IN
```

```
(SELECT nr_rejestr FROM naprawa GROUP BY nr_rejestr  
HAVING COUNT(*) >= ALL(SELECT COUNT(*) FROM naprawa GROUP BY nr_rejestr));
```

```
SELECT nazwa FROM firma WHERE id IN  
(SELECT firma_id FROM pojazd WHERE rok_prod < 2009 AND nr_rejestr IN  
(SELECT nr_rejestr FROM uslugi JOIN naprawa ON uslugi.id = uslugi_id  
WHERE nazwa = 'wymiana plynu chlodniczego'));
```