



EGZAMIN MATURALNY OD ROKU SZKOLNEGO 2014/2015

INFORMATYKA POZIOM ROZSZERZONY ARKUSZ I

PRZYKŁADOWY ZESTAW ZADAŃ (A1)

WYBRANE:

.....

(środowisko)

.....

(kompilator)

.....

(program użytkowy)

**Czas pracy:
60 minut**

GRUDZIEŃ 2013

Zadanie 1. (0–5)

W każdym z punktów od a) do e) zaznacz znakiem „x” poprawną odpowiedź.

a) Dana jest tabela *sprawdzian*:

<i>uczen</i>	<i>klasowka</i>	<i>egzamin</i>
Abacki	45	0
Babacki	50	80
Cabacki	100	90
Dabacki	80	70

Dla tej tabeli utworzono następujące zapytanie w SQL:

```
SELECT uczen
FROM sprawdzian
WHERE (klasowka > egzamin AND egzamin > 75) OR klasowka < 50
ORDER BY uczen
```

Wynikiem tego zapytania jest:

	Prawda	Fałsz
Abacki, Babacki		
Babacki, Cabacki		
Abacki, Cabacki		
Abacki, Dabacki		

b) Rozważmy poniższy algorytm, gdzie n jest liczbą całkowitą nieujemną:

- (1) $wynik \leftarrow 0$;
- (2) **dopóki** $n \neq 0$ **wykonuj**
- (3) $wynik \leftarrow wynik + (n \bmod 10)$
- (4) $n \leftarrow n \div 10$

oraz:

mod to operator reszty z dzielenia,

div to operator dzielenia całkowitego.

Dla podanego algorytmu zachodzi:

	Prawda	Fałsz
dla $n=36789$ $wynik=30$.		
dla $n=11111111$ $wynik=8$.		
$wynik$ jest równy sumie cyfr w zapisie dziesiętnym liczby n .		
dla $n=1234$ zmienna $wynik$ po kolejnych iteracjach pętli dopóki przyjmuje wartości 1, 3, 6, 10.		

c) Zgodnie z przepisami polskiego prawa autorskiego dozwolone jest:

	Prawda	Fałsz
publikowanie pod własnym nazwiskiem, na swojej stronie WWW, skopiowanych zasobów internetowych (zdjęć i artykułów).		
zamieszczanie na własnej stronie linków do innych stron WWW.		
zamieszczanie na własnej stronie cudzych programów na licencji freeware z podaniem ich autorstwa.		
zamieszczanie na stronie internetowej treści utworów wydanych wcześniej niż 70 lat temu.		

d) W grafice rastrowej

	Prawda	Fałsz
każdy piksel ma jednoznacznie określony kolor.		
obraz pamiętany jest w postaci obiektów geometrycznych.		
zaletą jest skalowalność obrazu.		
zapisywane są zdjęcia z aparatów cyfrowych.		

e) Dynamicznym przydzielaniem numerów IP w sieci zajmuje się serwer

	Prawda	Fałsz
DNS.		
DHCP.		
SMTP.		
FTP.		

Zadanie 2. (0–6)

Całkowity pierwiastek kwadratowy

Niech n będzie dodatnią liczbą całkowitą. **Całkowitym pierwiastkiem kwadratowym** z liczby n nazywamy dodatnią liczbę całkowitą k taką, że $k * k \leq n$ i $(k + 1) * (k + 1) > n$. Na przykład 3 jest całkowitym pierwiastkiem kwadratowym z liczb 9, 10, 11, 12, 13, 14 i 15. W tym zadaniu analizujemy algorytmy obliczania całkowitych pierwiastków z dodatnich liczb całkowitych, które mają być poprawne względem następującej specyfikacji:

Specyfikacja:

Dane: dodatnia liczba całkowita n

Wynik: dodatnia liczba całkowita k – całkowity pierwiastek kwadratowy z liczby n

Przykład: dla $n = 39$ wynikiem jest $k = 6$

a) W poniższym algorytmie uzupełnij instrukcję w wierszu (3) tak, żeby otrzymany algorytm był poprawny względem podanej wcześniej specyfikacji.

- (1) $k \leftarrow 1$;
- (2) **dopóki** $(k+1)*(k+1) \leq n$ **wykonuj**
- (3) $k \leftarrow \dots\dots\dots$;

Podaj, ile razy warunek w wierszu (2) powyższego algorytmu jest sprawdzany odpowiednio dla $n = 32$ i $n = 1024$.

n	<i>liczba sprawdzeń warunku w wierszu 2</i>
32	
1024	

Miejsce na obliczenia:

[illegible]

b) W poniższym algorytmie uzupełnij instrukcję w wierszu (5) tak, żeby otrzymany algorytm był poprawny względem podanej wcześniej specyfikacji.

- (1) $k \leftarrow 1; m \leftarrow n;$
- (2) **dopóki** $(k+1)^*(k+1) \leq n$ **wykonuj**
- (3) $s \leftarrow (k+m) \text{ div } 2;$
- (4) **jeśli** $s*s \leq n$ **to**
- (5) $k \leftarrow \dots\dots\dots$
- (6) **w przeciwnym przypadku**
- (7) $m \leftarrow s$

Uwaga: użyty operator div oznacza dzielenie całkowite, tzn. s jest największą liczbą całkowitą nie większą od $(k+m)/2$.

c) Podaj, ile razy warunek w wierszu (2) z algorytmu z punktu b) jest sprawdzany odpowiednio dla $n = 32$ i $n = 1024$.

n	<i>liczba sprawdzeń warunku w wierszu 2</i>
32	
1024	

[illegible]

Progi i schody

W ciągu liczb naturalnych, **schodami do dołu** nazywamy każdy jego podciąg kolejnych elementów, złożony z przynajmniej dwóch liczb, w którym każda liczba poza pierwszą nie jest większa od poprzedniej, a samego podciągu nie można rozszerzyć w żadną stronę do innych schodów do dołu. **Długością schodów do dołu** nazywamy liczbę zawartych w nim elementów.

Ciąg: 3, 7, 7, 6, 5, 4, 4, 4, 5 zawiera schody do dołu 7, 7, 6, 5, 4, 4, 4 o długości 7. Te schody zawierają 3 progi: pierwszy to 7 i 6, drugi to 6 i 5, trzeci to 5 i 4.

- [illegible]

b) Rozważmy następującą specyfikację:

Dane: dodatnia liczba całkowita n oraz tablica $a[1..n]$ zawierająca n -elementowy ciąg liczb całkowitych $a[1], \dots, a[n]$

Wynik: liczba całkowita *liczba_progów* – liczba wszystkich progów w ciągu zapisanym w tablicy *a*

W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) opracuj algorytm zgodny z powyższą specyfikacją.

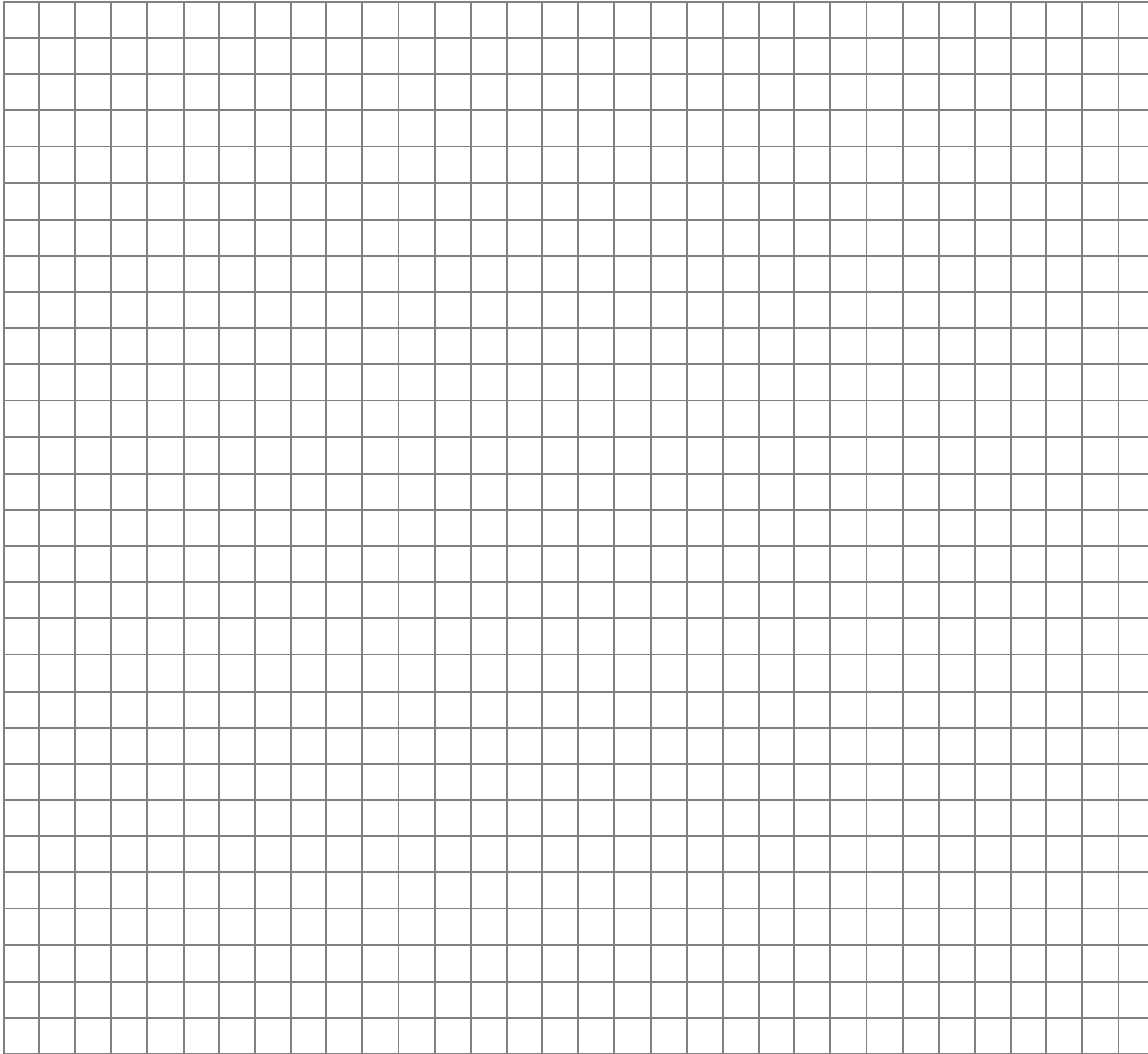
This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin, dark gray lines. There are 20 columns and 20 rows of squares, creating a total of 400 square units. The background is white, and the grid covers the entire area of the page.

c) Rozważmy następującą specyfikację:

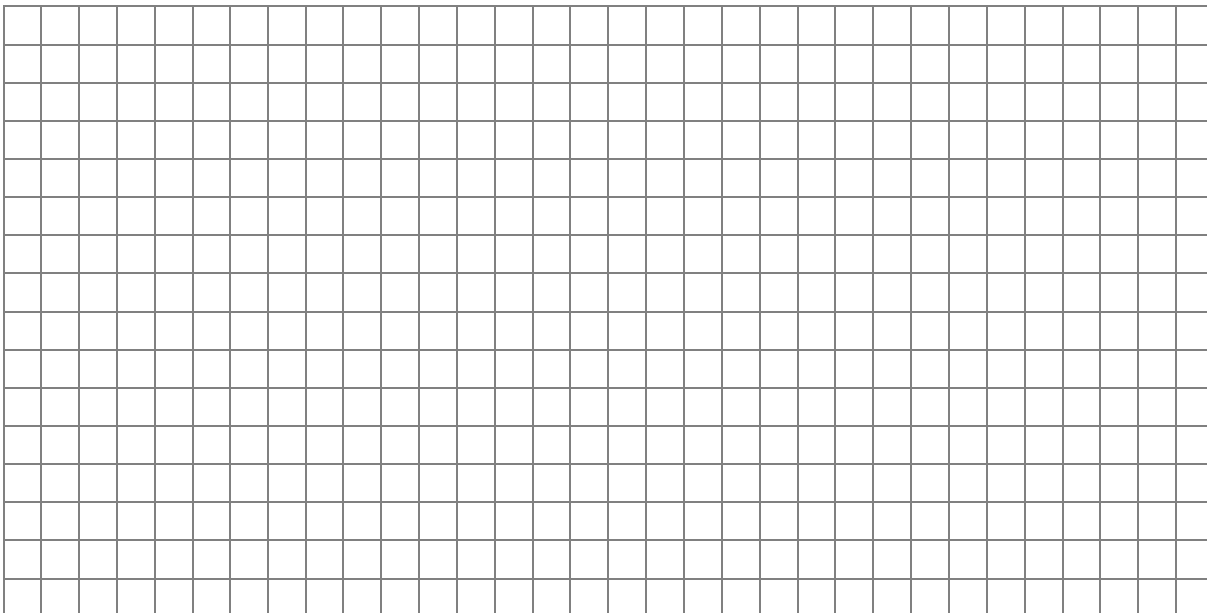
Dane: dodatnia liczba całkowita n oraz tablica $a[1..n]$ zawierająca n -elementowy ciąg liczb całkowitych $a[1], \dots, a[n]$

Wynik: liczba całkowita $najw_liczba_progów$ – największą liczbę progów w schodach do dołu z ciągu zapisanego w tablicy a

W wybranej przez siebie notacji (schemat blokowy, lista kroków, wybrany przez Ciebie język programowania) opracuj algorytm zgodny z powyższą specyfikacją.



- d) Podaj, ile dokładnie porównań między elementami ciągu danych wykona w pesymistycznym przypadku Twój algorytm z punktu c). Odpowiedź uzasadnij.



BRUDNOPIS (*nie podlega ocenie*)