

### Zadanie 1. Analiza algorytmu

Rozważamy następujący algorytm:

Dane:

$n$  – liczba całkowita dodatnia

Wynik:

$p$  – liczba całkowita dodatnia

```
p ← 1
q ← n
dopóki p < q wykonuj
    s ← (p+q) div 2
    (*) jeżeli s*s*s < n wykonaj
        p ← s+1
    w przeciwnym wypadku
        q ← s
```

Uwaga: zapis **div** oznacza dzielenie całkowite.

#### Zadanie 1.1. (0–3)

Podaj wynik działania algorytmu dla wskazanych w tabeli wartości  $n$ .

$n$	$p$
28	4
64	4
80	5

Miejsce na obliczenia.

//prosty pseudokod	//Rozpisanie pierwszego przykładu
<b>n</b> = ?	iteracja 1:
<b>p</b> = 1	while (1 < 28){
<b>q</b> = <b>n</b>	0 = (1 + 28) / 2
<b>s</b> = 0	if (14*14*14 < 28) {
	1 = 14 + 1 //jeszcze się nie wykona
	} else {
while ( <b>p</b> < <b>q</b> ){	28 = 14
<b>s</b> = ( <b>p</b> + <b>q</b> ) / 2	}
if ( <b>s</b> * <b>s</b> * <b>s</b> < <b>n</b> ) {	
<b>p</b> = <b>s</b> + 1	
} else {	
<b>q</b> = <b>s</b>	
}	
}	
	Kolejne Iteracje aż do wykonania się if'a
	i przzerwania się pętli

**Zadanie 1.2. (0–2)**

Podaj najmniejszą oraz największą liczbę  $n$ , dla której wynikiem działania algorytmu będzie  $p = 10$ .

Miejsce na obliczenia.

[illegible]

Odpowiedź: Najmniejsza liczba to **730**, największa liczba to **1000**.

### Zadanie 1.3. (0–1)

Dokończ zdanie. Wybierz i zaznacz właściwą odpowiedź spośród podanych.

Dla każdej liczby całkowitej  $n > 1$  instrukcja oznaczona w algorytmie symbolem (\*) wykona się

- A.** mniej niż  $2 \cdot \log_2 n$  razy.  
**B.** więcej niż  $n/2$ , ale mniej niż  $n$  razy.  
**C.** więcej niż  $n+1$ , ale mniej niż  $2n$  razy.  
**D.** więcej niż  $n^2$  razy.

<b>Wypełnia egzaminator</b>	<b>Nr zadania</b>	<b>1.1.</b>	<b>1.2.</b>	<b>1.3.</b>
	<b>Maks. liczba pkt.</b>	<b>3</b>	<b>2</b>	<b>1</b>
	<b>Uzyskana liczba pkt.</b>			

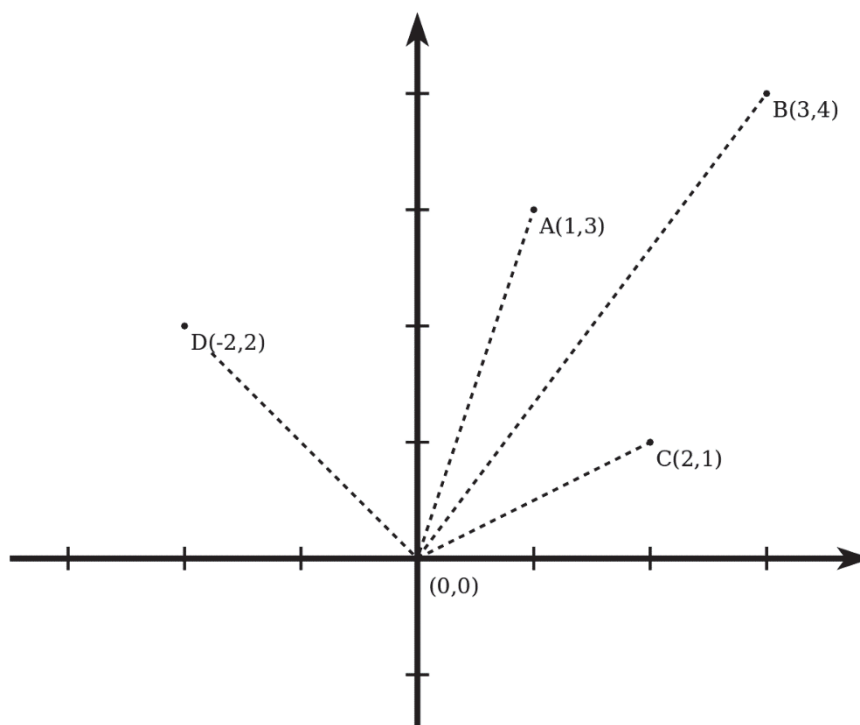
## Zadanie 2. Krajobraz

W pewnym paśmie górskim znajduje się  $n$  szczytów, które będziemy przedstawiać jako punkty w układzie kartezjańskim na płaszczyźnie. Wszystkie punkty leżą powyżej osi  $OX$ , tzn. druga współrzędna ( $y$ ) każdego punktu jest dodatnia.

W punkcie  $(0,0)$  stoi obserwator. Jeśli dwa szczyty  $A$  i  $B$  mają współrzędne  $(x_A, y_A)$  oraz  $(x_B, y_B)$ , to mówimy, że:

- szczyt  $A$  jest dla obserwatora *widoczny na lewo* od  $B$ , jeśli  $x_A/y_A < x_B/y_B$ ;
- szczyt  $B$  jest *widoczny na lewo* od  $A$ , jeśli  $x_A/y_A > x_B/y_B$ .

Wiemy, że żadne dwa szczyty nie leżą w jednej linii z obserwatorem, a zatem dla obserwatora te szczyty nie zasłaniają się nawzajem. Ilustrację przykładowego położenia szczytów można zobaczyć na poniższym rysunku:



W tym przykładzie, patrząc od lewej do prawej strony, obserwator widzi kolejno szczyt  $D$ , szczyt  $A$ , szczyt  $B$  i szczyt  $C$ .

Współrzędne szczytów dane są w dwóch tablicach  $X[1..n]$  oraz  $Y[1..n]$  – szczyt numer  $i$  ma współrzędne  $(X[i], Y[i])$ .

### Zadanie 2.1. (0–2)

Napisz algorytm (w pseudokodzie lub wybranym języku programowania), który znajdzie i poda współrzędne *skrajnie lewego szczytu*, tzn. widocznego dla obserwatora na lewo od wszystkich pozostałych szczytów.

## Specyfikacja:

*Dane:*

$n$  – liczba całkowita dodatnia

$X[1..n]$  – tablica liczb całkowitych

$Y[1..n]$  – tablica liczb całkowitych dodatnich

Para  $(X[i], Y[i])$  to współrzędne jednego szczytu,  $i = 1, 2, \dots, n$ .

Żadne dwa szczyty nie leżą w jednej linii z obserwatorem.

*Wynik:*

$x, y$  – współrzędne skrajnie lewego szczytu spośród tych opisanych w tablicach  $X$  i  $Y$ .

## Algorytm

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin gray lines. There are 20 columns and 20 rows of these squares, creating a total of 400 square units. The grid covers the entire area of the page, leaving no margins or other markings.

<b>Wypełnia egzaminator</b>	<b>Nr zadania</b>	<b>2.1.</b>
	<b>Maks. liczba pkt.</b>	<b>2</b>
	<b>Uzyskana liczba pkt.</b>	

### Zadanie 2.2. (0–4)

Napisz algorytm (w pseudokodzie lub wybranym języku programowania), który przestawi elementy tablic  $X$  i  $Y$  tak, aby szczyty były uporządkowane w kolejności, w której obserwator widzi je od lewej do prawej strony. Aby otrzymać maksymalną ocenę, Twój algorytm powinien mieć złożoność czasową kwadratową lub mniejszą.

Algorytm może używać wyłącznie instrukcji sterujących, operatorów arytmetycznych, operatorów logicznych, porównań i przypisań do zmiennych. Zabronione jest używanie funkcji bibliotecznych dostępnych w językach programowania.

## Specyfikacja:

*Dane:*

$n$  – liczba całkowita dodatnia

$X[1..n]$  – tablica liczb całkowitych

$Y[1..n]$  – tablica liczb całkowitych dodatnich

Para  $(X[i], Y[i])$  to współrzędne jednego szczytu,  $i = 1, 2, \dots, n$ .

Żadne dwa szczyty nie leżą w jednej linii z obserwatorem.

*Wynik:*

$X[1..n]$ ,  $Y[1..n]$  – tablice zawierające współrzędne danych szczytów, uporządkowanych w kolejności, w której obserwator widzi je od lewej do prawej strony.

## Algorytm

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 these squares, creating a total of 400 individual square units. The background is white, and the grid covers the entire area of the page without any margins or additional markings.

### Zadanie 3. Test

Oceń prawdziwość podanych zdań. Zaznacz P, jeśli zdanie jest prawdziwe, albo F – jeśli jest fałszywe.

W każdym zadaniu punkt uzyskasz tylko za komplet poprawnych odpowiedzi.

#### Zadanie 3.1. (0–1)

Na pewnym serwerze WWW znajduje się strona napisana w języku PHP, a jej kod zawiera fragmenty w języku JavaScript. Pewien komputer-klient pobrał i wyświetlił tę stronę. Wiadomo, że:

1.	kod PHP jest wykonywany przez komputer – serwer.	<input checked="" type="radio"/> P	F
2.	kod JavaScript jest wykonywany przez komputer – klient.	<input checked="" type="radio"/> P	F
3.	podczas wykonywania kodu PHP zawsze pobierane są dane od klienta.	P	<input checked="" type="radio"/> F
4.	podczas wykonywania kodu JavaScript mogą być pobierane dodatkowe dane zarówno od klienta, jak i od serwera.	<input checked="" type="radio"/> P	F

#### Zadanie 3.2. (0–1)

1.	Plakat do druku lepiej przygotować w modelu barw RGB niż CMYK.	P	<input checked="" type="radio"/> F
2.	Kolor żółty jest kolorem podstawowym w modelu RGB.	P	<input checked="" type="radio"/> F
3.	W wyniku nałożenia się składowych Yellow i Magenta w modelu CMYK otrzymamy kolor czerwony.	<input checked="" type="radio"/> P	F
4.	W modelu barw CMYK litera C pochodzi od angielskiego słowa <i>contrast</i> .	P	<input checked="" type="radio"/> F

#### Zadanie 3.3. (0–1)

Wskaż zdania prawdziwe dla języka SQL.

1.	W wynikach zapytania postaci <code>SELECT (...) ORDER BY (...)</code> zawsze dostajemy rekordy uporządkowane ściśle rosnąco według wskazanego pola.	P	<input checked="" type="radio"/> F
2.	Zapytanie <code>UPDATE</code> może zmienić wartości pól w bazie danych.	<input checked="" type="radio"/> P	F
3.	Zapytanie postaci <code>SELECT * FROM tabela1 WHERE pole LIKE (...)</code> może w pewnych warunkach dać wszystkie rekordy z tabeli <i>tabela1</i> .	<input checked="" type="radio"/> P	F
4.	Wynik zapytania <code>SELECT * FROM tabela1 JOIN tabela2 ON tabela1.pole = tabela2.pole</code> może być pusty przy niepustych tabelach <i>tabela1</i> oraz <i>tabela2</i> .	<input checked="" type="radio"/> P	F

Wypełnia egzaminator	Nr zadania	2.2.	3.1.	3.2.	3.3.
	Maks. liczba pkt.	4	1	1	1
	Uzyskana liczba pkt.				