



Sito Eratostenesa – zadania maturalne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Sito Eratostenesa – zadania maturalne

Źródło: Pixabay, domena publiczna.

Sito Eratostenesa jest algorytmem wyznaczającym liczby pierwsze z danego przedziału $[2, n]$. Liczby pierwsze znajdują zastosowanie m.in. w kryptografii, o czym przeczytasz w e-materiałach:

- [Zastosowanie liczb pierwszych](#),
- [Szyfr RSA](#).

W tym e-materiale znajdziesz zadania dotyczące implementacji tego algorytmu wraz z ich omówieniem.

Implementację omawianego algorytmu przedstawiamy w e-materiałach:

- [Sito Eratostenesa w języku C++](#),
- [Sito Eratostenesa w języku Java](#),
- [Sito Eratostenesa w języku Python](#).

Podstawowe informacje na jego temat znajdziesz w e-materiale [Sito Eratostenesa](#).

Twoje cele

- Przeanalizujesz rozwiązania zadań maturalnych dotyczących wyznaczania liczb pierwszych z danego przedziału.

- Rozwiążesz samodzielnie zadania maturalne, wykorzystując sito Eratostenesa.
- Prześledzisz schemat oceniania omawianych zadań maturalnych.

Przeczytaj

Przykładowe zadania maturalne

Zadanie 1. Pary

W pliku `pary.txt` znajduje się 100 wierszy. Każdy wiersz zawiera parę danych składającą się z liczby całkowitej z przedziału od 3 do 100 i słowa (ciągi znaków) złożone z małych liter alfabetu angielskiego o długości od 1 do 50 znaków. Liczba i słowo są oddzielone znakiem spacji.

Napisz program lub programy, będące odpowiedziami na podane zadania. Uzyskane odpowiedzi zapisz w pliku `wyniki4.txt`, poprzedzając każdą z nich numerem odpowiedniego zadania.

Uwaga: plik `przyklad.txt` zawiera przykładowe dane spełniające warunki zadania. Odpowiedzi dla danych z pliku `przyklad.txt` są podane pod treściami zadań oraz w pliku `odp_przyklad.txt`.

Plik `przyklad.txt` do zadania nr 1

Plik o rozmiarze 1 012.00 B w języku polskim

Plik `odp_przyklad.txt` do zadania nr 1

Plik o rozmiarze 865.00 B w języku polskim

Plik `pary.txt` do zadania nr 1

Plik o rozmiarze 1.71 KB w języku polskim

Zadanie 1.1.

Mocna hipoteza Goldbacha mówi, że każda parzysta liczba całkowita większa od 4 jest sumą **dwóch nieparzystych liczb pierwszych**, np. liczba 20 jest równa sumie $3 + 17$ lub sumie $7 + 13$.

Każdą **liczbę parzystą** z pliku `pary.txt` przedstaw w postaci sumy dwóch liczb pierwszych. Wypisz tę liczbę oraz dwa składniki sumy w kolejności niemalejącej. Jeżeli istnieje więcej rozwiązań (tak jak dla liczby 20), należy wypisać składniki sumy o największej różnicy.

Wyniki podaj w oddzielnych wierszach, w kolejności zgodnej z kolejnością danych w pliku `pary.txt`. Liczby w każdym wierszu rozdziel znakiem spacji, np. dla liczby 20 należy wypisać `20 3 17`.

Zadanie zostało przygotowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2020 roku (cz. 2.). Cały arkusz można

znaleźć na stronie internetowej CKE.

Ponieważ na egzaminie maturalnym uczniowie mają swobodę wyboru języka programowania, nie będziemy przedstawiać rozwiązania zadania w konkretnym języku, lecz posłużymy się pseudokodem.

Dokonaj implementacji proponowanego pseudokodu w języku, w którym programujesz.

Dla danych z pliku `przyklad.txt` prawidłową odpowiedzią jest:

```
1 24 5 19
2 6 3 3
3 6 3 3
```

Przykładowe rozwiązanie

Nasze rozwiązanie będzie opierać się na algorytmie sita Eratostenesa. Podczas omawiania rozwiązania skupimy się wyłącznie na części związanej z liczbami pierwszymi.

Rozwiązanie zadania rozpoczniemy od wczytania liczb z pliku. Zapisujemy je w tablicy `liczby`.

```
1 liczby = wczytaj_liczby_z_pliku
```

Następnie tworzymy tablicę wartości logicznych o nazwie `sito`, która posłuży do wskazywania liczb pierwszych. Z treści zadania wynika, że największą zapisaną w pliku liczbą jest 100 – nasza tablica będzie o 1 większa od największej możliwej liczby (tę wartość zapisujemy do zmiennej `n`). Każdej komórce w tablicy przypisujemy wartość `prawda` – początkowo zakładamy, że każda liczba jest pierwsza.

```
1 liczby = wczytaj_liczby_z_pliku
2 n = 101
3
4 sito[n] //tablica wartości logicznych
5
6 dla i = 0, 1, ... n wykonuj:
7     sito[i] = True
```

Kolejnym krokiem jest napisanie algorytmu sita Eratostenesa. Komórkom, których indeksy są liczbami złożonymi, przypiszemy wartość `fałsz`. Proces zmiany wartości nazywamy

„wykreślaniem”. Po wykonaniu tej części kodu w tablicy `sito`, indeksy w których komórkach pozostała wartość `prawda`, są liczbami pierwszymi.

```
1 liczby = wczytaj liczby z pliku
2 n = 101
3
4 sito[n] //tablica wartości logicznych
5
6 dla i = 0, 1, ... n wykonuj:
7     sito[i] = True
8
9 i = 2
10 dopóki i*i <= n wykonuj :
11     jeżeli sito[i] == prawda wykonaj:
12         j = i * i
13         dopóki j < n wykonuj:
14             sito[j] = fałsz
15             j = j + i
16     i = i + 1
```

W następnym kroku tworzymy dwie zmienne. Pierwsza z nich: `liczbaLiczbPierwszych` posłuży nam do zliczenia, ile liczb pierwszych udało nam się odszukać w badanym przedziale. Druga: `liczbyPierwsze` będzie tablicą liczb całkowitych – zapiszemy w niej znalezione liczby pierwsze.

```
1 liczby = wczytaj liczby z pliku
2 n = 101
3
4 sito[n] //tablica wartości logicznych
5
6 dla i = 0, 1, ... n wykonuj:
7     sito[i] = True
8
9 i = 2
10 dopóki i*i <= n wykonuj :
11     jeżeli sito[i] == prawda wykonaj:
12         j = i * i
13         dopóki j < n wykonuj:
14             sito[j] = fałsz
15             j = j + i
```

```

16     i = i + 1
17
18 liczbaLiczpPierwszych = 0
19 liczbyPierwsze = pustaTablica // pusta tablica liczb całkowitych
20
21 dla i = 2, 3, ... n wykonuj:
22     jeżeli sito[i] == prawda wykonaj:
23         liczbaLiczpPierwszych = liczbaLiczpPierwszych + 1
24         dodaj do tablicy liczbyPierwsze wartość i

```

Zakończyliśmy fazę przygotowania potrzebnych nam struktur danych. Nadszedł czas na przetworzenie liczb, które udało się pobrać z pliku. Piszemy pętlę dla, która będzie wskazywać kolejne liczby z tablicy `liczby`. Zgodnie z treścią polecenia, naszym zadaniem jest przedstawienie każdej liczby parzystej z pliku, przy pomocy sumy dwóch liczb pierwszych. Dopiszmy więc warunek badający parzystość przetwarzanej liczby. W warunku tym wykorzystamy operator `mod`, który służy do wyznaczania reszty z dzielenia (modulo). Dana liczba jest parzysta, jeżeli reszta z dzielenia tej liczby przez 2 wynosi 0.

```

1  liczby = wczytaj liczby z pliku
2  n = 101
3
4  sito[n] //tablica wartości logicznych
5
6  dla i = 0, 1, ... n wykonuj:
7      sito[i] = True
8
9  i = 2
10 dopóki i*i <= n wykonuj :
11     jeżeli sito[i] == prawda wykonaj:
12         j = i * i
13         dopóki j < n wykonuj:
14             sito[j] = fałsz
15             j = j + i
16     i = i + 1
17
18 liczbaLiczpPierwszych = 0
19 liczbyPierwsze = pustaTablica // pusta tablica liczb całkowitych
20
21 dla i = 2, 3, ... n wykonuj:
22     jeżeli sito[i] == prawda wykonaj:
23         liczbaLiczpPierwszych = liczbaLiczpPierwszych + 1

```

```

24         dodaj do tablicy liczbyPierwsze wartość i
25
26 dla i = 0, 1, ... n wykonuj:
27     jeżeli liczby[i] mod 2 == 0 wykonaj:

```

Wewnątrz instrukcji warunkowej tworzymy dwie zmienne pomocnicze: lewa i prawa. Będą one indeksami odwołującymi się do tablicy `liczbyPierwsze`. Zmienna `lewa` zostanie indeksem liczby wskazującej lewy składnik dodawania. Z kolei `prawa` będzie wskazywać prawy składnik. Autorzy zadania oczekują, że w przypadku, gdy liczbę można wyrazić przy pomocy więcej niż jednej sumy, wskażemy tę z największą różnicą. Dlatego zmiennej `lewa` przypisujemy wartość 0 (pierwszą – najmniejszą liczbę pierwszą z tablicy `liczbyPierwsze`). Z kolei zmiennej `prawa` – wartość `liczbaLiczbyPierwszych - 1` (czyli największą znaną liczbę pierwszą).

```

1  liczby = wczytaj liczby z pliku
2  n = 101
3
4  sito[n] //tablica wartości logicznych
5
6  dla i = 0, 1, ... n wykonuj:
7      sito[i] = True
8
9  i = 2
10 dopóki i*i <= n wykonuj :
11     jeżeli sito[i] == prawda wykonaj:
12         j = i * i
13         dopóki j < n wykonuj:
14             sito[j] = fałsz
15             j = j + i
16
17 liczbaLiczbyPierwszych = 0
18 liczbyPierwsze = pustaTablica // pusta tablica liczb całkowitych
19
20 dla i = 2, 3, ... n wykonuj:
21     jeżeli sito[i] == prawda wykonaj:
22         liczbaLiczbyPierwszych = liczbaLiczbyPierwszych + 1
23         dodaj do tablicy liczbyPierwsze wartość i
24
25 dla i = 0, 1, ... n wykonuj:
26     jeżeli liczby[i] mod 2 == 0 wykonaj:
27         lewa = 0

```

W pętli dopóki przeszukujemy znalezione liczby pierwsze – szukamy takich, które spełniają równanie

$\text{liczbyPierwsze}[\text{lewa}] + \text{liczbyPierwsze}[\text{prawa}] == \text{liczby}[i]$. Jeżeli wartość sumy jest mniejsza od $\text{liczby}[i]$, zwiększamy wartość zmiennej lewa. Z kolei jeżeli wartość sumy jest większa od $\text{liczby}[i]$, zmniejszamy wartość zmiennej prawa. Ostatnią operacją, którą chcemy wykonać, jest wypisanie wartości tworzących równanie.

```
1 liczby = wczytaj liczby z pliku
2 n = 101
3
4 sito[n] //tablica wartości logicznych
5
6 dla i = 0, 1, ... n wykonuj:
7     sito[i] = True
8
9 i = 2
10 dopóki i*i <= n wykonuj :
11     jeżeli sito[i] == prawda wykonaj:
12         j = i * i
13         dopóki j < n wykonuj:
14             sito[j] = fałsz
15             j = j + i
16     i = i + 1
17
18 liczbaLiczbPierwszych = 0
19 liczbyPierwsze = pustaTablica // pusta tablica liczb całkowitych
20
21 dla i = 2, 3, ... n wykonuj:
22     jeżeli sito[i] == prawda wykonaj:
23         liczbaLiczbPierwszych = liczbaLiczbPierwszych + 1
24         dodaj do tablicy liczbyPierwsze wartość i
25
26 dla i = 0, 1, ... n wykonuj:
27     jeżeli liczby[i] mod 2 == 0 wykonaj:
28         lewa = 0
29         prawa = liczbaLiczbPierwszych - 1
30         dopóki lewa <= prawa wykonuj:
31             jeżeli liczbyPierwsze[lewa] + liczbyPierwsze[prawa] =
32                 przerwij pętlę
```

```

33         w przeciwnym przypadku jeżeli liczbyPierwsze[lewa] +
34             lewa = lewa + 1
35         w przeciwnym przypadku:
36             prawa = prawa - 1
37     wypisz liczby[i], liczbyPierwsze[lewa], liczbyPierwsze[pr

```

Odpowiedź:

Plik "wynik.txt" do zadania nr 1

Plik o rozmiarze 247.00 B w języku polskim

Schemat oceniania:

Wymagania ogólne	Wymagania szczegółowe
Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.	Zdający analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin. Zdający stosuje podejście algorytmiczne do rozwiązywania problemu. Zdający formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera.

Zasady oceniania:

- **3 pkt** – za poprawną odpowiedź;
- **2 pkt** – za odpowiedź, w której jest podany dowolny poprawny rozkład na dwie liczby pierwsze dla każdej liczby parzystej ALBO gdy zdający nie uwzględni warunku, że liczba, której rozkładu szukamy, musi być parzysta i szuka także rozkładów liczb nieparzystych (ale rozkłady dla liczb parzystych otrzymuje poprawnie);
- **1 pkt** – za odpowiedź zawierającą rozkłady, w których dla każdej liczby parzystej występuje przynajmniej jedna liczba pierwsza większa od 3;
- **0 pkt** – za odpowiedź niepoprawną albo za brak odpowiedzi.

Schemat oceniania pochodzi z arkusza odpowiedzi egzaminu maturalnego z informatyki z 2020 roku. Cały arkusz można znaleźć na stronie internetowej Centralnej Komisji Egzaminacyjnej.

Słownik

kryptografia

dziedzina kryptologii, zajmująca się zabezpieczaniem informacji przed niepowołanym dostępem

liczba pierwsza

liczba naturalna większa od 1, która dzieli się tylko przez jeden i przez samą siebie

sito Eratostenesa

algorytm, który przesiewa liczby z określonego przedziału w taki sposób, że zostają w nim tylko liczby pierwsze; służy on do znajdowania wszystkich liczb pierwszych w podanym przedziale

Prezentacja multimedialna

Zadanie 2. Wybrane liczby

Liczby pierwsze to liczby naturalne większe od 1, które mają dokładnie dwa dzielniki: jedynkę i samą siebie.

Dane są dwa pliki: `liczby.txt` i `pierwsze.txt`. Plik `liczby.txt` zawiera 300 wierszy. W każdym wierszu tego pliku znajduje się jedna liczba całkowita dodatnia z zakresu od 1 do 100 000.

Plik `pierwsze.txt` zawiera 200 wierszy. W każdym wierszu tego pliku znajduje się jedna liczba pierwsza z zakresu od 10 do 1 300 000.

Uwaga: pomocnicze pliki `liczby_przyklad.txt` i `pierwsze_przyklad.txt` zawierają dane, które możesz wykorzystać, aby sprawdzić poprawność działania swoich programów. Każdy z plików zawiera po 50 wierszy. W każdym wierszu znajduje się jedna liczba. Odpowiedzi dla danych z tych plików są podane pod treściami zadań.

Plik `liczby_przyklad.txt` do zadania nr 2

Plik o rozmiarze 297.00 B w języku polskim

Plik `pierwsze_przyklad.txt` do zadania nr 2

Plik o rozmiarze 307.00 B w języku polskim

Napisz program lub programy, których działanie umożliwi otrzymanie odpowiedzi do podanych zadań. Pliki źródłowe z rozwiązaniem zapisz pod nazwą zgodną z numerem zadania, z rozszerzeniem odpowiadającym użytemu językowi programowania.

Plik `pierwsze.txt` do zadania nr 2

Plik o rozmiarze 1.29 KB w języku polskim

Zadanie 2.1.

Podaj (zachowując ich kolejność) te liczby z pliku `liczby.txt`, które są liczbami pierwszymi z przedziału $100; 5000$. Odpowiedź zapisz w pliku `wyniki2_1.txt`.

Dla pliku `liczby_przyklad.txt` odpowiedzią są liczby: 103, 163, 173, 701, 1033, 2137, 3529, 4933, 977, 2143.

Przykład:

Jeśli odczytamy liczbę pierwszą 17 od prawej do lewej, otrzymamy liczbę 71, która również jest liczbą pierwszą.

Zadanie zostało przygotowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2019 roku (cz. 2.).

Polecenie 1

Przedstaw rozwiązanie w wybranym języku programowania: C++, Java, Python. Odpowiedź dla danych z pliku znajdziesz pod prezentacją omawiającą kolejne kroki rozwiązania.

1

Polecenie 2

Zapoznaj się z prezentacją, w której omówiono kolejne kroki rozwiązania.

Ponieważ na egzaminie maturalnym uczniowie mają swobodę wyboru języka programowania, nie będziemy przedstawiać rozwiązania zadania w konkretnym języku, lecz posłużymy się pseudokodem.



Eratosthenes (276–194 r. p.n.e.) grecki matematyk, filozof, geograf, poeta i astronom, zasłynął między innymi tym, że podał sposób znajdowania liczb pierwszych znany pod nazwą Sito Eratostenesa.
Źródło: en.wikipedia.org, domena publiczna.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Podobnie jak w poprzednim zadaniu nasze rozwiązanie będzie opierać się na algorytmie sita Eratostenesa.

Implementacje rozpoczniemy od wczytania liczb z pliku. Zapisujemy je w tablicy pierwsze.

```
1  
2 pierwsze = wczytaj liczby  
3   z pliku
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Następnie tworzymy tablicę wartości logicznych o nazwie `sito`, która posłuży nam do wskazywania liczb pierwszych. Z treści zadania wynika, że największą zapisaną w pliku liczbą jest 1 300 000. Zauważmy, że będziemy badać nie największą liczbę w pliku, lecz jej zapis czytany od prawej do lewej. Zdecydujemy się na stworzenie dużo większego sita – jego rozmiar wyniesie 9 999 999 (tę wartość zapisujemy do zmiennej `n`). Każdej komórce w tablicy przypisujemy wartość `prawda` – początkowo zakładamy, że każda liczba jest pierwsza.

```
1
2 pierwsze = wczytaj liczby
  z pliku
3 n = 9 999 999
4
5 sito[n] //tablica wartości
  logicznych
6
7 dla i = 0, 1, ... n
  wykonuj:
8     sito[i] = True
9
10
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Podobnie jak podczas prezentacji rozwiązania poprzedniego zadania, tworzymy sito Eratostenesa. Komórkom, których indeksy są liczbami złożonymi, przypiszemy wartość `fałsz`. Wartość `prawda` pozostanie

niezmieniona w komórkach, których indeksy są liczbami pierwszymi.

```
1
2 pierwsze = wczytaj liczby
  z pliku
3 n = 9 999 999
4
5 sito[n] //tablica wartości
  logicznych
6
7 dla i = 0, 1, ... n
  wykonuj:
8     sito[i] = True
9
10 i = 2
11 dopóki i*i <= n wykonuj :
12     jeżeli sito[i] ==
  prawda wykonaj:
13         j = i * i
14         dopóki j < n
  wykonuj:
15             sito[j] =
  fałsz
16             j = j + i
17     i = i + 1
18
19
```

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Przetwarzamy kolejne liczby zapisane w tablicy pierwsze. Wewnątrz pętli dla tworzymy dwie zmienne pomocnicze: `liczba` oraz `odwroconaLiczba`. Pierwszej przypisujemy aktualnie przetwarzaną liczbę – wartość `pierwsze[i]`, drugą inicjalizujemy wartością 0.

```

1
2 pierwsze = wczytaj liczby
  z pliku
3 n = 9 999 999
4
5 sito[n] //tablica wartości
  logicznych
6
7 dla i = 0, 1, ... n
  wykonuj:
8     sito[i] = True
9
10 i = 2
11 dopóki i*i <= n wykonuj :
12     jeżeli sito[i] ==
  prawda wykonaj:
13         j = i * i
14         dopóki j < n
  wykonuj:
15             sito[j] =
  fałsz
16             j = j + i
17     i = i + 1
18
19 dla i = 0, 1, 2 ... 200
  wykonuj:
20     liczba = pierwsze[i]
21     odwroconaLiczba = 0
22
23

```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Następnie odwracamy liczbę. Piszemy pętlę dopóki, która będzie wykonywać się do czasu, aż liczba jest większa od 0. Wewnątrz pętli zapisujemy do zmiennej bieżącaCyfra cyfrę jedności liczby. Uzyskujemy ją, wykonując działanie `liczba mod 10`. Operator mod

wyznacza resztę z dzielenia, a cyfra jedności liczby jest po prostu resztą z dzielenia tej liczby przez 10. "Przesuwamy" dotychczasowe cyfry zapisane w zmiennej `odwroconaLiczba`, mnożąc jej wartość przez 10, a także dodajemy do niej wartość zmiennej `biezacaCyfra`. Wartość zmiennej `liczba` dzielimy całkowitoliczbowo przez 10, uzyskując tym samym liczbę zmniejszoną o wartość jedności. Operator `div` pozwala nam uzyskać wynik takiego dzielenia, z pominięciem reszty.

```
1
2 pierwsze = wczytaj liczby
  z pliku
3 n = 9 999 999
4
5 sito[n] //tablica wartości
  logicznych
6
7 dla i = 0, 1, ... n
  wykonuj:
8     sito[i] = True
9
10 i = 2
11 dopóki i*i <= n wykonuj :
12     jeżeli sito[i] ==
  prawda wykonaj:
13         j = i * i
14         dopóki j < n
  wykonuj:
15             sito[j] =
  fałsz
16             j = j + i
17     i = i + 1
18
19 dla i = 0, 1, 2 ... 200
  wykonuj:
20     liczba = pierwsze[i]
21     odwroconaLiczba = 0
22
23     dopóki liczba > 0
  wykonuj:
24         biezacaCyfra =
  liczba mod 10
```

```

25         odwroconaLiczba =
(odwroconaLiczba * 10) +
biezacaCyfra
26         liczba = liczba
div 10
27
28

```

6



Eratostenes znany jest również z tego, że obliczył obwód Ziemi oraz oszacował jej odległość od Słońca.

Źródło: Pixabay, CC 0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PcqgBS9sW>

Jeżeli w `sito` pod indeksem `odwroconaLiczba` zapisano wartość `prawda` oznacza to, że liczba jest pierwsza. Drukujemy wartość `pierwsze[i]`.

```

1
2 pierwsze = wczytaj liczby
z pliku
3 n = 9 999 999
4

```

```

5 sito[n] //tablica wartości
  logicznych
6
7 dla i = 0, 1, ... n
  wykonuj:
8     sito[i] = True
9
10 i = 2
11 dopóki i*i <= n wykonuj :
12     jeżeli sito[i] ==
prawda wykonaj:
13         j = i * i
14         dopóki j < n
wykonuj:
15             sito[j] =
fałsz
16             j = j + i
17     i = i + 1
18
19 dla i = 0, 1, 2 ... 200
  wykonuj:
20     liczba = pierwsze[i]
21     odwroconaLiczba = 0
22
23     dopóki liczba > 0
wykonuj:
24         biezacaCyfra =
liczba mod 10
25         odwroconaLiczba =
(odwroconaLiczba * 10) +
biezacaCyfra
26         liczba = liczba
div 10
27
28     jeżeli
sito[odwroconaLiczba] ==
prawda:
29         wypisz pierwsze[i]
30

```

Odpowiedź:

Plik "wyniki2_1.txt" do zadania nr 2.1

Plik o rozmiarze 324.00 B w języku polskim

Schemat oceniania:

Wymagania ogólne	Wymagania szczegółowe
Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.	Zdający analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin. Zdający stosuje podejście algorytmiczne do rozwiązywania problemu. Zdający formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera.

Zasady oceniania:

- **4 pkt** – za poprawną odpowiedź (wypisanie wszystkich 51 liczb);
- **3 pkt** – za pominięcie jednej liczby;
- **2 pkt** – za pominięcie dwóch liczb;
- **1 pkt** – za pominięcie trzech liczb;
- **0 pkt** – za pominięcie czterech i więcej liczb.

Schemat oceniania pochodzi z arkusza odpowiedzi egzaminu maturalnego z informatyki z czerwca 2019 roku.

Polecenie 3

Dodaj do swojego programu komentarze tak, żeby był zrozumiały dla osoby, która nie potrafi programować.

Sprawdź się

Zadanie 3. Punkty XY

Plik punkty.txt zawiera 1000 wierszy, w których zapisane są po dwie liczby całkowite z przedziału $[0, 10000]$, oddzielone pojedynczym odstępem. Para liczb w każdym wierszu odpowiada współrzędnym (x, y) jednego punktu w układzie kartezjańskim. W danych punkty się nie powtarzają.

Napisz program lub programy, umożliwiające uzyskanie odpowiedzi do przedstawionych zadań. Odpowiedzi zapisz w pliku wyniki4.txt, a każdą odpowiedź poprzedź numerem oznaczającym odpowiednie zadanie.

Plik punkty.txt do zadania nr 3

Plik o rozmiarze 10.53 KB w języku polskim

Zadanie 3.1.

Ile jest punktów, których **obie współrzędne** są liczbami pierwszymi?

Specyfikacja:

Dane:

x, y – liczby naturalne z zakresu $[0, 10000]$ opisujące współrzędne punktu w układzie kartezjańskim

Wynik:

Na standardowym wyjściu program prezentuje liczbę punktów, których obie współrzędne są liczbami pierwszymi.

Zadanie zostało przygotowane przez Centralną Komisję Egzaminacyjną i pojawiło się na egzaminie maturalnym z informatyki w czerwcu 2017 roku (cz. 2.).

Pamiętaj o uruchomieniu programu w środowisku umożliwiającym otwieranie plików.

Pokaż ćwiczenia:   

Ćwiczenie 1



Implementacja w języku C++

Ćwiczenie 2



Implementacja w języku Java



Implementacja w języku Python

Odpowiedź:

1 18

Schemat oceniania:

Wymagania ogólne	Wymagania szczegółowe
Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego.	Zdający analizuje, modeluje i rozwiązuje sytuacje problemowe z różnych dziedzin. Zdający stosuje podejście algorytmiczne do rozwiązywania problemu. Zdający formułuje przykłady sytuacji problemowych, których rozwiązanie wymaga podejścia algorytmicznego i użycia komputera.

Zasady oceniania:

- **2 pkt** – za poprawną odpowiedź (18);
- **1 pkt** – podanie wyniku 236, który oznacza liczbę punktów, których przynajmniej jedna współrzędna jest liczbą pierwszą;
- **0 pkt** – za wynik różny od 18 i 236.

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Sito Eratostenesa – zadania maturalne

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

I + II. Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) zapisuje za pomocą listy kroków, schematu blokowego lub pseudokodu, i implementuje w wybranym języku programowania, algorytmy poznane na wcześniejszych etapach oraz algorytmy:

c) generowania liczb pierwszych metodą sita Eratostenesa,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Przeanalizujesz rozwiązania zadań maturalnych dotyczących wyznaczania liczb pierwszych z zadanego przedziału.
- Rozwiążesz samodzielnie zadania maturalne, wykorzystując sito Eratostenesa.
- Prześledzisz schemat oceniania omawianych zadań maturalnych.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio;
- oprogramowanie dla języka Java SE 8 (lub nowszej wersji), w tym Eclipse 4.4 (lub nowszej wersji);
- oprogramowanie dla języka Python 3 (lub nowszej wersji), w tym PyCharm lub IDLE.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Sito Eratostenesa – zadania maturalne”. Nauczyciel prosi uczniów o zapoznanie się z multimedium w sekcji „Prezentacja multimedialna”.

Faza wstępna:

1. Przedstawienie tematu i celów zajęć.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel zadaje uczniom pytania dotyczące ich aktualnego stanu wiedzy w obszarze poruszanego tematu, np.
 - do czego służy tzw. sito Eratostenesa?
 - czym charakteryzują się liczby pierwsze?Chętni lub wybrani uczniowie udzielają na nie odpowiedzi.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimedium.** Nauczyciel czyta polecenie nr 1 z sekcji „Prezentacja multimedialna”. Prosi uczniów, aby wykonali je w parach. Następnie wybrana osoba prezentuje propozycję odpowiedzi, a pozostali uczniowie ustosunkowują się do niej. Nauczyciel w razie potrzeby uzupełnia ją, udziela też uczniom informacji zwrotnej.
3. Uczniowie rozwiązują ćwiczenia 1–3 z sekcji „Sprawdź się”. Nauczyciel dzieli klasę na grupy, zależnie od wybranego języka programowania. Uczniowie w grupach omawiają napisane przez siebie programy.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie proponują alternatywny sposób rozwiązania problemów postawionych w sekcji „Przeczytaj”.
2. Uczniowie wykonują polecenie 3 z sekcji „Prezentacja multimedialna”.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Uczniowie mogą wykorzystać multimedium w sekcji „Prezentacja multimedialna” do przygotowania się do lekcji powtórkowej.