

Algorytm Sito Eratostenesa – C++ Step by step

Badanie własności liczb całkowitych można wykorzystać na kilka sposobów. Można badać parzystość i nieparzystość liczby, jej podzielność przez jakąś liczbę, czy sprawdzać czy jest to liczba pierwsza, czy posiada liczbę bliźniaczą, itp... .

Ćwiczenie.: Sprawdzenie czy liczba jest parzysta

Z matematycznego punktu widzenia można powiedzieć, że liczba jest parzysta jeżeli jest podzielna przez liczbę 2. W celu sprawdzenia czy liczba jest parzysta musimy określić, czy liczba po podzieleniu jest liczbą całkowitą. Można to zrobić badając resztę z dzielenia. Musi ona wynosić zero i wówczas liczba jest liczbą parzystą, w przeciwnym wypadku nie jest to liczba parzysta.

Przykład algorytmu w C++ sprawdzającego parzystość:

```
*main.cpp x
1      #include <iostream>
2      using namespace std;
3
4      int main()
5      {
6          int x;
7          cout << "Podaj liczbe do sprawdzenia:" << endl;
8          cin >> x; //wczytanie liczby z klawiatury
9          if(x%2==0){ //if sprawdza warunek parzystości
10             cout << "Liczba jest parzysta";
11         }
12         else {
13             cout << "To nie jest liczba parzysta" << endl;
14         }
15         return 0;
16     }
```

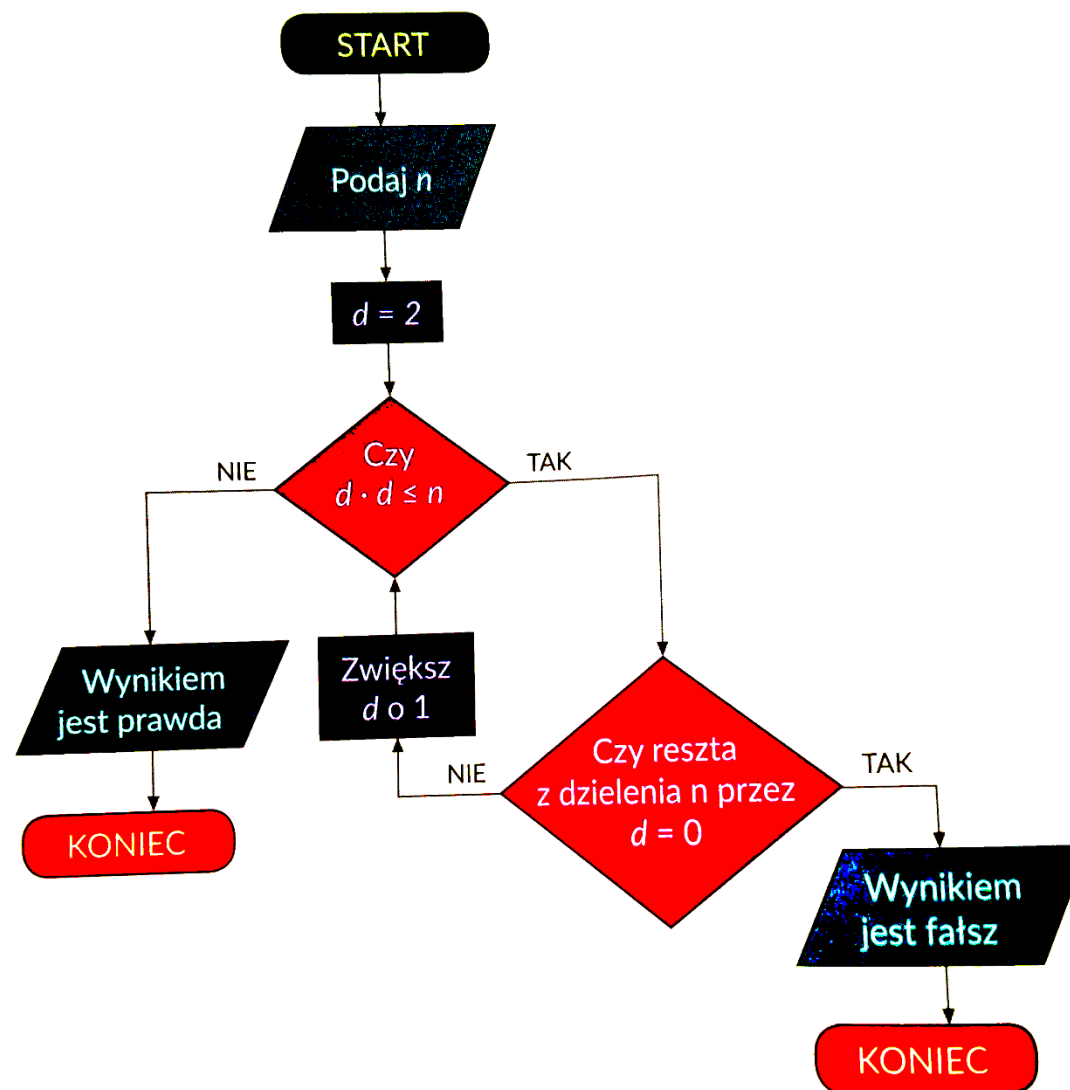
Co to jest sito Eratostenesa?

Jest to tablica z wykreślanymi (odsianymi) wielokrotnościami liczb niewykreślonych idąc od najmniejszej. Liczby od której zaczynamy nie możemy wykreślić. Liczby, które zostaną nie skreślone są liczbami pierwszymi. Przykładowe liczby pierwsze:

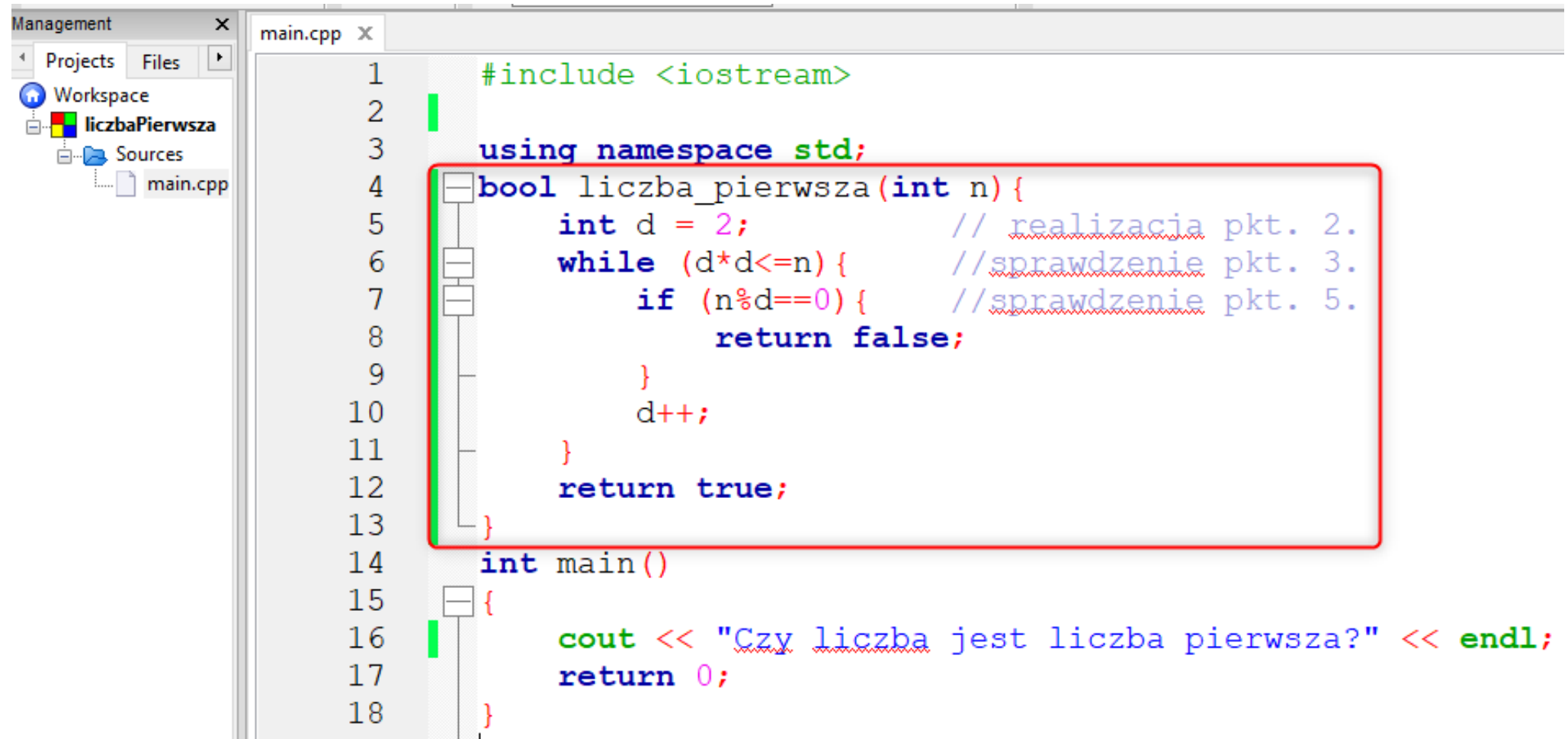
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

W jaki sposób wyznaczyć liczby pierwsze? Algorytm:

1. Wczytaj liczbę 'n'
2. Wpisz do zmiennej 'd' wartość '2'
3. Sprawdź, czy jest spełniony warunek: $d \cdot d \leq n$
4. Jeżeli powyższy warunek **NIE** jest spełniony to liczba jest liczbą pierwszą
5. Jeżeli powyższy warunek jest spełniony to sprawdź wynik reszty z dzielenia liczby n przez d
6. Jeżeli reszta z powyższego dzielenia jest równa zero to nie jest to liczba pierwsza. Zwiększ d o 1 i przejdź do punktu 3.
7. Jeżeli reszta z powyższego dzielenia nie jest równa zero do liczby d dodaj 1 i przejdź do punktu 3.



Algorytm w C++ sprawdzający czy liczba jest liczbą pierwszą. W pierwszym kroku tworzymy funkcję boolowską o nazwie liczba_pierwsza:



```
1  #include <iostream>
2
3  using namespace std;
4  bool liczba_pierwsza(int n) {
5      int d = 2;           // realizacja pkt. 2.
6      while (d*d<=n) {     // sprawdzenie pkt. 3.
7          if (n%d==0) {    // sprawdzenie pkt. 5.
8              return false;
9          }
10         d++;
11     }
12     return true;
13 }
14 int main()
15 {
16     cout << "Czy liczba jest liczba pierwsza?" << endl;
17     return 0;
18 }
```

The screenshot shows a C++ IDE with a project named 'liczbaPierwsza'. The file 'main.cpp' is open, displaying the code for a prime number checker. The function 'liczba_pierwsza' is highlighted with a red box. The code includes the necessary headers, uses the 'std' namespace, and implements a while loop to check for divisibility. The 'main' function prints the question and returns 0.

Powyższa funkcja jest funkcją Boolowską sprawdzającą czy liczba jest pierwszą. Jeżeli pętla WHILE trafi na liczbę nie będącą liczbą pierwszą to wartość FALSE kończy pętlę. Pętla jest również sprawdzana do momentu znalezienia liczby pierwszej. Wówczas funkcja zwraca wartość TRUE reprezentowaną w C++ przez jedynkę. Jedynka ta jest przekazywana do programu głównego MAIN.

Teraz trzeba stworzyć program przechodzący przez wszystkie badane liczby z zakresu od 2 do n i wypisujące liczby pierwsze. Przykładowy program może mieć postać:

```
*main.cpp x
13  }
14  int main()
15  {
16      cout << "Czy liczba jest liczba pierwsza? Podaj gorna liczba:" << endl;
17      int n; //zmienna górnego zakresu liczb badanych
18      cin >> n; //wczytanie 'n' z klawiatury
19      int i=2; //pierwsza z 'n' sprawdzanych liczb
20      int k=1; //zmienna pomocnicza przechodząca przez 'n' liczb
21      while(k != n){ //sprawdzenie warunku czy osiągnęliśmy 'n'
22          k++;
23          if(liczba_pierwsza(i)==1){ //czy jest to liczba pierwsza
24              cout << i << " "; //wyświetlenie liczby pierwszej
25          }
26          i++; //zwiększamy licznik dot. badanych liczb
27      }
28      return 0;
29  }
```

Sprawdź działanie powyższego programu u siebie.