

**student:** Imię Nazwisko (GRUPA)  
student@elka.pw.edu.pl  
**prowadzący:** Imię Nazwisko

## Podstawy Programowania (PRM) — projekt specyfikacja strukturalna (techniczna)

**Temat:** Znajdowanie wszystkich liczb pierwszych zawartych w pewnym przedziale.

### Algorytm rozwiązania zadania

Do rozwiązania zadania wykorzystana będzie metoda sita Eratostenesa. W metodzie tej, rozważane są kolejne liczby od 2 do ograniczenia górnego  $N$ . Każda liczba z przedziału  $[2, N]$  ma status:

- 0 do rozważenia
- 1 pierwsza
- 2 nie pierwsza

Algorytm działa w sposób następujący:

```
procedura sito Eratostenesa
dane wejściowe  $M$  (ograniczenie dolne),  $N$  (ograniczenie górne)
{
  dla każdego  $i \in [2, N]$ 
     $status(i) = 0$ 
  dla każdego  $i \in [2, N]$ 
    jeśli  $status(i) == 0$ 
      dla każdego  $j \in [i, N]$ 
         $status(j) = 2$ 
  dla każdego  $i \in [M, N]$ 
    jeśli  $status(i) == 1$ 
       $wypisz(i)$ 
}
```

### Struktury danych

Program będzie się posługiwał tablicą zawierającą  $(N - M + 1)$  elementów, w których będzie przechowywana wartość statusu liczby. Element o indeksie  $i$  (numeracja od zera) będzie odpowiadał liczbie  $i+M$ .

### Podział na funkcje

- `int sito(char status[], int M, int N);`

Funkcja wykonująca obliczenia wg algorytmu Eratostenesa. Przyjmuje następujące argumenty

- `char status[]` Tablica zawierająca wartości statusu poszczególnych liczb; konwencja numeracji elementów tablicy opisana wyżej.
- `int M, int N` Ograniczenia: dolne i górne przedziału poszukiwań

Wartością zwracaną jest liczba różnych liczb pierwszych z zakresu  $[M, N]$  (0 oznacza brak).

- `int czytNat(char* zacheta);`

Funkcja wyprowadza na standardowe wyjście napis będący argumentem `char* zacheta`, a następnie wczytuje kolejne łańcuchy znakowe ze standardowego wejścia, dopóki nie dadzą się zinterpretować jako liczba naturalna w formacie dziesiętnym. Wartość tej liczby jest zwracana jako wynik funkcji.

- `int przygotuj(int M, int N);`

Funkcja przygotowująca dane dla funkcji `sito`. Przechowuje tablicę zawierającą wartości statusu. Wykonywane są kolejno następujące czynności:

1. jeśli wartości  $M, N$  są ujemne, są one wczytywane funkcją `czytNat`; jeśli tylko  $M$  lub tylko  $N$  jest ujemna, użytkownik jest proszony o podanie brakującej wartości;
2. ewentualna zamiana wartości  $M, N$ , jeśli  $M > N$ ;
3. alokacja tablicy `status` o  $M - N + 1$  elementach; w przypadku niepowodzenia, funkcja wyprowadza na standardowy strumień błędów stosowny komunikat i zwraca wartość -1;
4. wywołanie funkcji `sito`;
5. w zależności od wyniku, wypisanie komunikatu o braku rozwiązań lub wypisanie rozwiązań;
6. zwolnienie pamięci z tablicy `status`;
7. zwrócenie wartości 0.

- `int main(int argc, char* argv[]);`

Funkcja główna. Wywołuje funkcję `przygotuj`:

1. jeśli program został wywołany bez argumentów, przekazywane są wartości  $M, N$  równe -1
2. jeśli został podany jeden argument wywołania, jest on wartością  $M$ , a wartością  $N$  jest -1;
3. jeśli zostały podane dwa argumenty lub więcej, pierwsze dwa są traktowane jako wartości  $M, N$ , a pozostałe są odrzucane;
4. w przypadku gdy argument nie daje się zinterpretować jako liczba całkowita, przyjmowana jest jego wartość równa -1.

Zwraca wartość zwracaną przez `przygotuj`.

## Literatura

1. Wirth, N. *Algorytmy + struktury danych = programy*.
2. Kernighan, B., Ritchie, D. *Język ANSI C*
3. Silvester, P. *System operacyjny unix*.