

Laboratorium 1: Algorytmy z pętlami

1. Algorytm Euklidesa

1.1. Wyznacz największy wspólny dzielnik dwóch liczb naturalnych za pomocą algorytmu Euklidesa

1.2. Algorytm:

- 1) Podaj liczbę a
- 2) Podaj liczbę b
- 3) Wybierz liczbę większą z liczb a i b
- 4) Przypisz liczbę nie mniejszą do $x1$ oraz pozostałą liczbę do $x2$
- 5) Ustaw $i:=0$
- 6) Wyznacz resztę dzielenia $x3:=x1 \bmod x2$
- 7) Jeśli $x3>0$, wykonuj kolejne kroki, w przeciwnym wypadku przejdź do kroku 8
- 7.1) Powiększ $i:=i+1$
- 7.2) Umieść $x3$ w tablicy w wierszu i oraz kolumnie 0
- 7.3) Wyznacz: $x1:=x2$ $x2:=x3$ $x3:=x1 \bmod x2$ i przejdź do kroku 7
- 8) $x2$ jest największym wspólnym dzielnikiem – wyświetl go na ekranie
- 9) wyświetl zawartość tablicy na ekranie
- 9.1) ustaw $j:=1$
- 9.2) dopóki $j<=i$ wykonuj kolejne kroki, w przeciwnym wypadku zakończ program
- 9.3) pobierz do zmiennej a element tablicy z wiersza j i kolumny 0
- 9.4) wyświetl a na ekranie
- 9.5) zwiększ $j:=j+1$ i przejdź do kroku 9.2

Nazwa	Wartość	Plansza	Pozio
a	11	euklides	1
b	3	euklides	1
c	1	euklides	1
i	2	euklides	1
j	3	euklides	1
x1	2	euklides	1
x2	1	euklides	1
x3	0	euklides	1

Plansza	Pozio	Wynik	Opis
euklides	1	Falsz	$x3 < 0$
euklides	1	1	największy wspólny dzielnik
euklides	1	1	$i = 1$
euklides	1	Prawda	$j < = i$
euklides	1	2	Odczyt z tablicy: <0> <j>
euklides	1	2	kolejna reszta z dzielenia
euklides	1	2	$i = i + 1$
euklides	1	Prawda	$j < = i$
euklides	1	1	Odczyt z tablicy: <0> <j>
euklides	1	1	kolejna reszta z dzielenia
euklides	1	3	$i = i + 1$
euklides	1	Falsz	$j < = i$
euklides	1		Koniec algorytmu

Operacja	Liczba
Dodawanie	0
Odejmowanie	0
Mnożenie	0
Dzielenie	0
Moduł	0
Przypisanie	0
Porównanie	0
Wywołanie procedury	0
Funk. trygonometryczne	0
Funk. logarytmiczne	0
Inne funkcje	0
Odwołania do tablicy	0
Odwołania do taśmy	0

2. Algorytm wyszukania liczb pierwszych metodą sita Eratostenesa

2.1. Należy wyznaczyć wszystkie liczby pierwsze w podziorze liczb naturalnych $\{1..N\}$ za pomocą algorytmu sita Eratostenesa

2.2. Algorytm:

- 1) Utworzyć tablicę zawierającą N elementów i wstawić do każdego elementu wartość 0
 - 1.1) Podaj N z klawiatury
 - 1.2) Jeśli $N < 2$, powtórz krok 1.1
 - 1.3) Ustaw $i := 2$
 - 1.4) Dopóki $i \leq N$ wykonuj kolejne kroki, w przeciwnym wypadku przejdź do kroku 2**
 - 1.4.1) wstaw 0 do elementu tablicy o indeksie i
 - 1.4.2) zwiększ $i := i + 1$ i przejdź do kroku 1.4.

- 2) Zakłada się, że pewne indeksy elementów są szukanymi liczbami pierwszymi i po zakończeniu algorytmu elementy tablicy o tych indeksach będą zawierać wartość 0, natomiast pozostałe elementy mają wartość 1, ponieważ nie są liczbami pierwszymi. Stąd należy wstawić na początku wartość 1 do elementu o indeksie równym 1, ponieważ 1 nie jest liczbą pierwszą
- 3) Ustawić $ost_Liczp:=1$;
- 4) Na podstawie faktu, że każda liczba złożona nie większa niż N ma dzielnik nie większy niż \sqrt{N} wykonuj kolejne kroki, gdy $ost_Liczp*ost_Liczp \leq N$, w przeciwnym wypadku przejdź do kroku 4.6:
 - 4.1) Należy zwiększyć ost_Liczp o 1: $ost_Liczp:=ost_Liczp+1$
 - 4.2) Wykonuj kolejne kroki, jeśli jest prawdziwy warunek ($ost_Liczp \leq N$) and ($tab[ost_Liczp]=1$), w przeciwnym wypadku przejdź do kroku 4.3.
 - 4.2.1) zwiększaj ost_Liczp o 1: $ost_Liczp:=ost_Liczp+1$ (poszukiwanie kolejnej liczby pierwszej, czyli elementu tablicy o indeksie ost_Liczp nie zawierającej wartości 1)
 - 4.4.2) przejdź do kroku 4.2
 - 4.3) Należy podwoić ost_Liczp $ost_i:=ost_Liczp*2$ (rozpoczęcie kolejnego etapu wykreślenia liczb, które nie są liczbami pierwszymi)
 - 4.4) Dopóki $i \leq N$, wykonaj w kolejnych krokach eliminację liczb, które nie są liczbami pierwszymi, ponieważ są ich wielokrotnościami, w przeciwnym wypadku przejdź do kroku 4.
 - 4.4.1) wstaw wartość 1 to elementu tablicy o wierszu równym i
 - 4.4.2) dodaj wartość ost_Liczp do i : $i:=i+ost_Liczp$, następnie przejdź do kroku 4.4
 - 4.6) Wyświetl zawartość tablicy na ekranie:
 - 4.6.1) wstaw $i:=1$
 - 4.6.2) dopóki $i \leq N$ wykonuj kolejne kroki, w przeciwnym wypadku zakończ algorytm
 - 4.6.2.1) jeśli $tab[i]=0$, wyświetl indeks elementu jako wartość kolejnej liczby pierwszej
 - 4.6.2.2) wyznacz kolejny indeks $i:=i+1$ i przejdź do kroku 4.6.2.

The screenshot shows the Elbox Laboratory Informatics 2.01 software interface. The main window displays a flowchart for the Sieve of Eratosthenes algorithm. The flowchart starts with a loop for i from 1 to N . For each i , it checks if $tab[i]=0$. If true, it prints i and marks multiples of i as 1. If false, it increments i and repeats the loop. The flowchart ends when $i > N$.

Below the flowchart, there is a table showing the complexity of operations:

Operacja	Liczba
Dodawanie	20
Odejmowanie	0
Mnożenie	7
Dzielenie	0
Moduł	0
Przypisanie	25
Porównanie	32
Wywołanie procedury	0
Funk. trygonometryczne	0
Funk. logarytmiczne	0
Inne funkcje	0
Odwwołania do tablicy	21
Odwwołania do taśmy	0

On the right side, there is a 'Ślad' (Trace) window showing the execution steps:

Opis	0	1
0		
1	1	
2	0	
3	0	
4	1	
5	0	
6	1	
7	0	
8	1	
9	1	
10	1	

Below the trace, there is a 'Zmienne' (Variables) window showing the current values of variables:

Nazwa	Wartość	Plansza	Pozior
N	10	sito	1
a	1	sito	1
i	11	sito	1
ost_Liczp	5	sito	1