

# Wstęp do Informatyki i Programowania

## Laboratorium nr 4

20, 26 listopada i 2, 3, 11 grudnia

### Zadanie 1C (6 pkt)

Napisz w języku C, na podstawie poprzednich list, bibliotekę zawierającą następujące funkcje

- `unsigned prime_numbers(unsigned n)` - zwracającą liczbę liczb pierwszych mniejszych równych  $n$ ;
- `unsigned prime(unsigned n)` - zwracającą  $n$ -tą liczbę pierwszą;
- `bool is_prime(unsigned n)` - sprawdzającą czy podana liczba jest liczbą pierwszą.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał komendę i liczbę i zwracał na ekranie odpowiedni wynik. Komendy to odpowiednio `pn`, `pr` i `ip`. Przykładowe użycie skompilowanego pliku `program` powinno wyglądać następująco:

```
komputer$ ./program pn 100
25
komputer$ ./program pr 25
97
komputer$ ./program ip 100
false
komputer$ ./program ip 97
true
```

*Wskazówka: pamiętaj, że w C nie ma typu string tylko tablice typu char, dlatego do operacji na napisach należy użyć biblioteki `string.h`.*

### Zadanie 1A (6 pkt)

Napisz w języku Ada, na podstawie poprzednich list, pakiet zawierający następujące funkcje

- `function PrimeNumbers (n : Positive) return Positive` - zwracającą liczbę liczb pierwszych mniejszych równych  $n$ ;
- `function Prime(n : Positive) return Positive` - zwracającą  $n$ -tą liczbę pierwszą;
- `function IsPrime(n : Positive) return Boolean` - sprawdzającą czy podana liczba jest liczbą pierwszą.

Zadbaj aby poza tymi trzema funkcjami reszta funkcji i typów danych była prywatna.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał komendę i liczbę i zwracał na ekranie odpowiedni wynik. Komendy to odpowiednio `pn`, `pr` i `ip`. Przykładowe użycie skompilowanego pliku `program` powinno wyglądać następująco:

```
komputer$ ./program pn 100
25
komputer$ ./program pr 25
97
```

```
komputer$ ./program ip 100
False
komputer$ ./program ip 97
True
```

### Zadanie 1P (6 pkt)

Napisz w języku Python, na podstawie poprzednich list, pakiet zawierający następujące funkcje

- def PrimeNumbers( $n$ ) - zwracającą liczbę liczb pierwszych mniejszych równych  $n$ ;
- def Prime( $n$ ) - zwracającą  $n$ -tą liczbę pierwszą;
- def IsPrime( $n$ ) - sprawdzającą czy podana liczba jest liczbą pierwszą.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał komendę i liczbę i zwracał na ekranie odpowiedni wynik. Komendy to odpowiednio pn, pr i ip. Przykładowe użycie skompilowanego pliku program powinno wyglądać następująco:

```
komputer$ python program pn 100
25
komputer$ python program pr 25
97
komputer$ python program ip 100
False
komputer$ python program ip 97
True
```

### Zadanie 2C (4 pkt)

Napisz w języku C bibliotekę zawierającą następującą funkcję

- unsigned totient(unsigned  $n$ ) - zwracającą liczbę liczb względnie pierwszych mniejszych równych  $n$ ;

Wykorzystaj wcześniej napisany algorytm rozkładu liczby na czynniki pierwsze.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał ciąg liczb i zwracał na ekranie odpowiednie wyniki. Przykładowe użycie skompilowanego pliku program powinno wyglądać następująco:

```
komputer$ ./program 1 11 100 5000 12345678
totient(1) = 1
totient(11) = 10
totient(100) = 40
totient(5000) = 2000
totient(12345678) = 4027392
```

### Zadanie 2A (4 pkt)

Napisz w języku Ada pakiet zawierający następującą funkcję

- function Totient( $n$  : Positive) return Positive - zwracającą liczbę liczb względnie pierwszych mniejszych równych  $n$ ;

Wykorzystaj wcześniej napisany algorytm rozkładu liczby na czynniki pierwsze. Zadbaj aby poza tą funkcją reszta funkcji i typów danych była prywatna.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał ciąg liczb i zwracał na ekranie odpowiednie wyniki. Przykładowe użycie skompilowanego pliku program powinno wyglądać następująco:

```
komputer$ ./program 1 11 100 5000 12345678
totient(1) = 1
totient(11) = 10
totient(100) = 40
totient(5000) = 2000
totient(12345678) = 4027392
```

### **Zadanie 2P (4 pkt)**

Napisz w języku Python pakiet zawierający następującą funkcję

- def Totient(n) - zwracającą liczbę liczb względnie pierwszych mniejszych równych  $n$ ;

Wykorzystaj wcześniej napisany algorytm rozkładu liczby na czynniki pierwsze.

Następnie napisz program główny, który z parametrów linii poleceń będzie czytał ciąg liczb i zwracał na ekranie odpowiednie wyniki. Przykładowe użycie skompilowanego pliku program powinno wyglądać następująco:

```
komputer$ python program 1 11 100 5000 12345678
totient(1) = 1
totient(11) = 10
totient(100) = 40
totient(5000) = 2000
totient(12345678) = 4027392
```