Programowanie

Ćwiczenia, semestr letni 2023

Michał Suchorowski

20 marca 2023

Spis treści

1 Ćwiczenia 2 **1**
1.1 Zadanie 1 . . . 1
1.2 Zadanie 2 . . . 1
1.3 Zadanie 3 . . . 1
1.4 Zadanie 4 . . . 1
1.5 Zadanie 5 . . . 1
1.6 Zadanie dodatkowe 1 . . . 3
1.7 Zadanie dodatkowe 2 . . . 3
1.8 Zadanie dodatkowe 3 . . . 3
1.9 Zadanie dodatkowe 4 . . . 3
1.10 Zadanie dodatkowe 5 . . . 3
1.11 Zadanie dodatkowe 6 . . . 3

1 Ćwiczenia 2

1.1 Zadanie 1

Wypisz liczby naturalne od 0 n (n wczytaj z klawiatury). Wypisz je na 3 sposoby używając pętli **while**, **for** oraz **do while**. Kolejne liczby powinny być wypisane w tym samym wierszu. Wynik z każdego rodzaju pętli w kolejnych wierszach. Użyj operatora ++.

1.2 Zadanie 2

Wypisz w trzech kolumnach liczby od 0 do n (n wczytane z klawiatury), gdzie w pierwszej kolumnie będzie n, w drugiej n^2, a w trzeciej suma $\sum_m^n m^2$. Użyj operatora +=. Użyj formatowania tekstu include<iomanip>, **setw**, aby ustawić odstęp między kolumnami.

1.3 Zadanie 3

Napisz program, który wczytuje liczby a i b, a następnie wypisuje a/b. Jeśli $b = 0$ to program ma poprosić o wpisanie o wpisanie prawidłowych liczb i wrócić do wpisywania liczb.

1.4 Zadanie 4

Sprawdź liczby cyfr znaczacych dla typow **float** i **double** poprzez liczenie sumy wyrazow $1/(10^i)$ od 0 do n. Użyj opcji **setprecision(20)**, żeby wyświetlić wszystkie liczby znaczące. (Zobacz jaki będzie wynik bez tej opcji, czyli jaka jest domyślna precyzja wyświetlania.)

1.5 Zadanie 5

Wczytaj liczbę i sprawdź czy jest pierwsza dla liczb z zakresu od 2 do 1e7. Wczytuj liczby dopóki wczytana liczba jest równa 0. Napisz program na dwa sposoby:
1) *brute force method*, czyli pętla, która sprawdza wszystkie możliwe dzielniki, 2) sito Eratostenesa (zobacz https://pl.wikipedia.org/wiki/Sito_Eratostenesa)

Napisz oba sposoby w oddzielnych programach, a potem przetestuj dla 50-100 wybranych liczb używając komendy Linuxa *time* (zapiszcie te liczb w tablicy w kodzie, żeby nie musieć wpisywać ich z klawiatury). Czas, który nas interesuje to *user*.

Przetestuj program zmieniając go, żeby sprawdzał czy liczby z tablicy T są pierwsze.

```
  int T[100]={0};
  for(int i=0; i<40; i++)
    T[i]=range-1;
```

```
    time ./a.out
```

Metoda **brute force**

```
#include <iostream>
#include<cmath>

using namespace std;

int main(){

  int a;
  int flag=0;

  int T[100]={0};
  for(int i=0; i<100; i++)
    T[i]=range-1;


   for(int j=0;j<100;j++ {
    flag=0;
//    cin >> a ;
    a=T[j];
    for(int i=(int)sqrt(a)+1; i>=2; --i){
        if(a%i==0){
            flag=1;
            continue;
        }
    }
    if(flag) cout << "Liczba nie jest pierwsza" << endl;
    else cout << "Liczba jest pierwsza" << endl;

    }

  return 0;
}
```

Metoda **sita**

W kwestii optymalizacji: wielokrotności liczby pierwszej, którym przypisujemy wartość 1 w tablicy muszą być sprawdzane do wartości maksymalnej **range** (a nie do pierwiastka z **range** jak w brute force!). Za to możemy zaczynać sprawdzanie od **j**=i (czyli od od kwadratu liczby), bo niższe wielokrotności już zostały wpisane do tablicy przy sprawdzaniu mniejszych liczb. Zobacie na animację tutaj, żeby się przekonować lub spróbujcie rozpisać to na kartce dla małych liczb: https://pl.wikipedia.org/wiki/Sito_Eratostenesa#/media/Plik:Sieve_of_Eratosthenes_animation.gif.

```
#include <iostream>
#include<cmath>

using namespace std;

#define range 10000000

int main(){


  bool sito[range+1]={0};

  for(int i=2; i<=range; i++){
    if(sito[i]==0){
        int j=i;
        while(j*i<range){
            sito[i*j]=1;
            j++;
        }
    }
  }

 cout << "sito policzone" << endl;

  int a;
  int T[100]={0};
  for(int i=0; i<100; i++)
    T[i]=range-1;
```

```
   for(int j=0;j<100;j++ {
  //   cin >> a ;
     a=T[j];

     if(sito[a]==0) cout << "Liczba jest pierwsza" << endl;
     else cout << "Liczba nie jest pierwsza" << endl;
     }

  return 0;
}
```

1.6 Zadanie dodatkowe 1

Przypomnienie z instrukcji warunkowej. Napisz program, który sprawdzi czy wczytany rok jest przestępny. Sprawdzamy czy rok został wpisany prawidłowo (nie akceptujemy ujemnych lat).

1.7 Zadanie dodatkowe 2

Napisz program, który policzy wartość $n!$. Sprawdź jak dużą silnie możemy uzyskać używając zmiennej typu **int**, a jaką **long long int**.

1.8 Zadanie dodatkowe 3

Wypisz tabliczkę mnożenia od 1 do n w formacie:

*	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

1.9 Zadanie dodatkowe 4

Zaimplementuj algorytm Euklidesa do znajdowania największego wspólnego dzielnika. Zobacz https://pl.wikipedia.org/wiki/Algorytm_Euklidesa.

1.10 Zadanie dodatkowe 5

Zlicz ile liczb w danym przedziale od 0 do n jest podzielne przez liczbę m. Liczbę n i m wczytaj z klawiatury.

1.11 Zadanie dodatkowe 6

Wypisz kolejne n wyrazów ciągu Fibonacciego.