

Zadanie 1

Napisz program wczytujący liczby całkowite aż do pojawienia się zera i wypisujący długość najdłuższej sekwencji kolejnych liczb o takich samych wartościach (i tę wartość). Na przykład dla

2 2 2 11 11 11 11 2 2 3 3 2 2 12 12

wynik powinien być Najdluzsza sekwencja: 4 razy liczba 11.

Nie używaj tablic, napisów ani żadnych innych kolekcji.

Zadanie 2

Napisz zestaw funkcji do badania liczb pierwszych i dzielników liczb.

- `bool isPrime(int n)` sprawdza, czy podana liczba jest pierwsza;
- `void printDivisors(int n)` wypisuje wszystkie dzielniki podanej liczby naturalnej (łącznie z jedynką i samą liczbą);
- `int numDivisors(int n)` zwraca liczbę dzielników podanej jako argument liczby naturalnej N (łącznie z jedynką i samą liczbą N);
- `int gcd(int m, int n)` znajduje największy wspólny dzielnik podanych dwóch liczb używając algorytmu Euklidesa (Księga VII i X *Elementów*):

Dla danych $a, b \in N$:

- Dopóki $a \neq b$: większą z liczb a, b pomniejsz o wartość mniejszej
- Zwróć a

- `int phi(int n)` zwraca dla danej liczby naturalnej N ilość wszystkich liczb z zakresu $[1, N)$ względnie pierwszych z N (dla których największy wspólny dzielnik z N wynosi 1). Jest to słynna funkcja φ Eulera. Na przykład $\varphi(10) = 4$, bo tyle jest liczb naturalnych mniejszych od 10 i względnie pierwszych z 10 (są to liczby 1, 3, 7 i 9).

Zadanie 3

Napisz funkcję o prototypie

```
const double* aver(const double* arr,
                   size_t size, double& average);
```

która pobiera tablicę liczb typu `double` i jej wymiar oraz zmienną `average` typu `double` przez referencję. Zadaniem funkcji jest wstawienie do zmiennej `average` średniej arytmetycznej elementów tablicy oraz zwrócenie wskaźnika do tego elementu tablicy, którego wartość jest najbliższa tej średniej.

Na przykład następujący program

download [AverClosest.cpp](#)

```

#include <iostream>
#include <cmath>

const double* aver(const double* arr,
                   size_t size, double& average) {
    // ...
}

int main () {
    using std::cout; using std::endl;

    double arr[] = {1,7,5,4,3,2};
    size_t size = sizeof(arr)/sizeof(arr[0]);
    double average = 0;
    const double* p = aver(arr, size, average);
    cout << *p << " " << average << endl;
}

```

powinien wypisać

```
4 3.66667
```

Zadanie 4

Napisz funkcję

```

void minMaxRep(int a[], size_t size, int& mn, size_t& in,
               int& mx, size_t& ix);

```

która pobiera tablicę **int**ów **a**, jej wymiar **size** oraz, przez referencje, cztery zmienne, do których wpisany ma być wynik działania funkcji: **mn**, **in**, **mx** i **ix**. Funkcja znajduje wartości najmniejszego i największego elementu tablicy i wpisuje je do zmiennych **mn** i **mx**, a do **in** i **ix** wpisuje, odpowiednio, liczbę wystąpień tej najmniejszej i największej wartości w całej tablicy.

Na przykład program

```

#include <iostream>
download MinMaxRepF.cpp

void minMaxRep(int a[], size_t size,
               int& mn, size_t& in, int&mx, size_t& ix) {
    // ...
}

int main() {
    using std::cout;
    int a[]{2,3,4,2,7,4,7,2};
    size_t size = sizeof(a)/sizeof(*a);
    int mn, mx;

```

```

    size_t in, ix;
    minMaxRep(a,size,mn,in,mx,ix);
    cout << "Array: [ ";
    for (size_t i = 0; i < size; ++i)
        cout << a[i] << " ";
    cout << "]\n";
    cout << "Min = " << mn << " " << in << " times\n";
    cout << "Max = " << mx << " " << ix << " times\n";
}

```

powinien wydrukować

```

Array: [ 2 3 4 2 7 4 7 2 ]
Min = 2 3 times
Max = 7 2 times

```

Uwaga: nie wolno stosować żadnych dodatkowych tablic ani kolekcji. Funkcja może przebiec w pętli po elementach tablicy tylko raz.
