

Zadanie 1

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 (co do modułu) najbliższa tej średniej.

Na przykład następujący program

```
#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[] = {1,2,3,4,5,7};
    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 2

Napisz program, który wczytuje w pętli ciąg liczb całkowitych do momentu, gdy użytkownik poda liczbę 0, która jest tylko sygnałem końca danych i nie jest dalej brana pod uwagę. Następnie program wypisuje wartości najmniejszego i największego elementu wczytanego ciągu oraz liczbę wystąpień tych wartości w całym ciągu.

Na przykład dla ciągu (2, 3, 4, 2, 7, 4, 7, 2) program powinien wypisać

```
Min = 2 3 razy
Max = 7 2 razy
```

Uwaga: nie wolno stosować żadnych tablic ani innych kolekcji!

Zadanie 3

Napisz funkcję pobierającą tablicę nieujemnych liczb całkowitych (i jej wymiar) i wypisującą na ekranie, za pomocą symboli '*' lub '=', „histogram” dla danych z tablicy, to znaczy „słupki”, wyrównane od dołu, o wysokościach równych wartościom kolejnych elementów tablicy.

Na przykład program

```
#include <iostream>
using namespace std;

void histo(int arr[], size_t size) {
    // ...
}

int main() {
    int arr[] {2,1,0,7,1,9};
    size_t size = sizeof(arr)/sizeof(*arr);
    histo(arr,size);
}
```

powinien wydrukować:

```
      *
      *
    * *
    * *
    * *
    * *
    * *
  * * *
** ***
```

Wybór symbolu ('*' lub '=') powinien być uzależniony od zdefiniowanego makra preprocesora. Na przykład, jeśli zdefiniowane jest makro STAR, to symbolem tym powinien być znak '*', a jeśli zdefiniowane jest makro EQU, to znak '=' (jeśli żadne z tych makr nie jest zdefiniowane, albo zdefiniowane są obydwa, to program nie powinien się w ogóle skompilować).

Zadanie 4

Napisz funkcję

```
size_t monot(const int a[], size_t size, size_t& len);
```

która w podanej tablicy *a* znajduje najdłuższy podciąg monotoniczny — nierosnący lub niemalejący (niekoniecznie *ściśle* monotoniczny). Funkcja zwraca indeks, od którego rozpoczyna się znaleziony podciąg, a do przekazanej przez referencję zmiennej

len wstawia długość tego podciągu. Jeśli jest kilka takich najdłuższych podciągów o tej samej długości, to zwracana jest informacja o pierwszym z nich.

Uwaga: w funkcji można użyć tylko *jednego* przebiegu pętli, bez pętli zagnieżdżonych i bez tworzenia pomocniczych tablic lub innych kolekcji.

Na przykład poniższy program (w którym funkcja **printResult** jest tylko pomocniczą funkcją wypisującą rezultat)

```
download ArrMonotonic.cpp
```

```
#include <iostream>

size_t monot(const int a[], size_t size, size_t& len) {
    // ...
}

void printResult(const int a[], size_t size,
                 size_t ind, size_t len) {
    using std::cout;
    cout << "In array [ ";
    for (size_t i = 0; i < size; ++i) cout << a[i] << " ";
    cout << "]\nthe first longest "
         << "monotonic sequence is\n[ ";
    for (size_t i = ind; i < ind+len; ++i)
        cout << a[i] << " ";
    cout << "]\nstarting at index " << ind
         << " and of length " << len << "\n\n";
}

int main() {
    size_t len = 0, ind = 0, size = 0;

    int a1[] = {4,4,5,4,4};
    size = sizeof(a1)/sizeof(*a1);
    ind = monot(a1,size,len);
    printResult(a1,size,ind,len);

    int a2[] = {4,4,5,4,4,4};
    size = sizeof(a2)/sizeof(*a2);
    ind = monot(a2,size,len);
    printResult(a2,size,ind,len);

    int a3[] = {4,5,4,4,5};
    size = sizeof(a3)/sizeof(*a3);
    ind = monot(a3,size,len);
    printResult(a3,size,ind,len);
}
```

powinien wypisać

In array [4 4 5 4 4]
the first longest monotonic sequence is
[4 4 5] starting at index 0 and of length 3

In array [4 4 5 4 4 4]
the first longest monotonic sequence is
[5 4 4 4] starting at index 2 and of length 4

In array [4 5 4 4 5]
the first longest monotonic sequence is
[5 4 4] starting at index 1 and of length 3
