

Zadanie 1

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>

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

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

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

Zadanie 2

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 3

Napisz funkcję o prototypie

```
double* aver(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 następująca funkcja **main**

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

powinna wypisać

```
4 3.66667
```

Zadanie 4

Napisz *przeciążone* funkcje

```
void ord3(double& a, double& b, double& c);
void ord3(double* a, double* b, double* c);
```

które pobierają trzy liczby typu **double** odpowiednio przez referencje i przez wskaźniki i porządkują je w kolejności wzrastającej (tak, że *po wyjściu* z funkcji ich wartości są zmienione).

Napisz *przeciążone* funkcje

```
void getMinMax(double &a, double& b, double& c,
               double*& ptrMin, double*& ptrMax);
void getMinMax(double *a, double* b, double* c,
               double** ptrMin, double** ptrMax);
```

które pobierają trzy liczby typu **double** odpowiednio przez referencje i przez wskaźniki i do wskaźników **ptrMin** i **ptrMax** przekazanych przez, odpowiednio, referencje i wskaźniki, wstawiają adresy zmiennych odpowiadających najmniejszej i największej z przekazanych liczb. Wartości przekazanych liczb nie ulegają zmianie.

Uwaga: Nie używaj tablic, napisów ani żadnych kolekcji. Funkcje nie mogą niczego pisać na ekran — wszystkie wyniki są drukowane w funkcji **main**.

Następujący program, bez żadnych zmian w funkcji **main**,

```
#include <iostream>
#include <utility> // you can use std::swap

void getMinMax(double &a, double& b, double& c,
               double*& ptrMin, double*& ptrMax) {
    // ...
}

void getMinMax(double *a, double* b, double* c,
               double** ptrMin, double** ptrMax) {
    // ...
}
```

download *RefsPointers.cpp*

```

}

void ord3(double& a, double& b, double& c) {
    // ...
}

void ord3(double* a, double* b, double* c) {
    // ...
}

int main() {
    using std::cout; using std::endl;
    double a, b, c, *ptrMin, *ptrMax;

    a = 2; b = 1; c = 3;
    ord3(a, b, c);
    cout << a << " " << b << " " << c << endl;

    a = 3; b = 2; c = 1;
    ord3(&a, &b, &c);
    cout << a << " " << b << " " << c << endl;

    a = 2; b = 3; c = 1; ptrMin = ptrMax = nullptr;
    getMinMax(a, b, c, ptrMin, ptrMax);
    std::cout << "Min = " << *ptrMin << "; "
               << "Max = " << *ptrMax << std::endl;

    a = 3; b = 1; c = 2; ptrMin = ptrMax = nullptr;
    getMinMax(&a, &b, &c, &ptrMin, &ptrMax);
    std::cout << "Min = " << *ptrMin << "; "
               << "Max = " << *ptrMax << std::endl;
}

```

powinien wypisać

```

1 2 3
1 2 3
Min = 1; Max = 3
Min = 1; Max = 3

```

Sprawdź program również dla innych danych; w szczególności dla sytuacji, gdy niektóre lub wszystkie wartości są równe.
