

## 1. Код

```
#define _CRT_SECURE_NO_WARNINGS

#define MAX_N 100

#include <stdio.h>
#include <iostream>

using namespace std;

void inputArray(int& n, double* A) {
    double tmp;
    n = 0;
    while (true) {
        cin >> tmp;
        if (tmp < 0.0) {
            break;
        }
        else {
            A[n++] = tmp;
        }
    }
}

void mergeArrays(int n, double* A, double* B, double* C) {
    for (int i = 0; i < n; i += 2) {
        C[i] = A[i];
    }
    for (int i = 1; i < n; i += 2) {
        C[i] = B[i];
    }
}

double calculateProduct(int n, double* A) {
    //returns -1.0 if there is no zero in array
    //otherwise returns product
    double product = 1.0;
    int lastZero = -1;
    for (int i = 0; i < n; i++) {
        if (A[i] == 0.0) {
            lastZero = i;
        }
    }
    if ((lastZero == -1) || (lastZero == n - 1)) {
        //no zero in array or no elements after last zero
        return -1.0;
    }
    for (int i = lastZero + 1; i < n; i++) {
        product *= A[i];
    }
    return product;
}

void printResult(int n, double* A, string arrayName) {
```

```

    cout << "array " << arrayName << ":\n";
    for (int i = 0; i < n; i++) {
        cout << A[i] << " ";
    }
    if (calculateProduct(n, A) == -1.0) {
        cout << endl << "cannot calculate the product\n\n";
    }
    else {
        cout << endl << "product: " << calculateProduct(n, A) << "\n\n";
    }
}

int main()
{
    int n;
    double A[MAX_N];
    double B[MAX_N];
    double C[MAX_N];
    cout << "enter two arrays of non-negative real numbers. negative element indicates
end of the input\n";
    inputArray(n, A);
    inputArray(n, B);

    mergeArrays(n, A, B, C);

    printResult(n, A, "A");
    printResult(n, B, "B");
    printResult(n, C, "C");

    return 0;
}

```

2.

Проверим работу программы на каком-нибудь входе. Пусть ввод следующий:

```
1 2 3 4 5 6 7 8 9 -10
1 0 2 0 3 3 3 3 3 -1.1
```

Вывод будет:

```
array A:
1 2 3 4 5 6 7 8 9
cannot calculate the prouct
```

```
array B:
1 0 2 0 3 3 3 3 3
product: 243
```

```
array C:
1 0 3 0 5 3 7 3 9
product: 2835
```

Произведение в А не считается, т.к. нет ни одного нуля, соответственно нет чисел после нуля и нет их произведения.

3.

## Несколько скринов вывода

```
Microsoft Visual Studio Debug Console
enter two arrays of non-negative real numbers. negative element indicates end of the input
1 2 3 4 5 -1
0 0 0 1 0 -2
array A:
1 2 3 4 5
cannot calculate the prouct

array B:
0 0 0 1 0
cannot calculate the prouct

array C:
1 0 3 1 5
product: 15

C:\Users\wicirelllis\source\repos\lab_2\Debug\lab_2.exe (process 13420) exited with code 0.
Press any key to close this window . . .
```

```
Microsoft Visual Studio Debug Console
enter two arrays of non-negative real numbers. negative element indicates end of the input
1 0 2 3 4 5 -1
1 2 3 4 0 6 -3
array A:
1 0 2 3 4 5
product: 120

array B:
1 2 3 4 0 6
product: 6

array C:
1 2 2 4 4 6
cannot calculate the prouct

C:\Users\wicirelllis\source\repos\lab_2\Debug\lab_2.exe (process 5672) exited with code 0.
Press any key to close this window . . .
```

## 4. Вопросы

1) Как хранятся в памяти элементы массива? Какими способами можно обратиться к элементам массива?

Элементы массива хранятся в памяти последовательно, друг за другом, одним куском. Т.е. соседние элементы массива лежат в соседних ячейках памяти.

Два распространенных метода обратиться к  $i$ -му элементу массива  $A$  это

$A[i]$   
 $*(A + i)$

Это работает потому что  $A$  – это не только имя массива, но и указатель (=адрес в памяти) на первый элемент. Например,  $A+1$  это адрес рядом с  $A$ , и в нем, соответственно, лежит первый элемент массива.  $*$  - это разыменовывание указателя.

Важная деталь, скобки нужны. Без них сначала произойдет разыменовывание, а потом  $+i$  и вместо  $A[i]$  получится  $A[0]+i$ .

2) Приведите пример кода, в котором объявляется функция, которая должна принимать массив целочисленных значений, дробное значение и возвращать булеву переменную.

```
bool myFunction(int myArrayLength, int* myArray, float myFloat);
```

3) Что будет выведено в консоль, при выполнении следующего кода:

```
int k = 5;  
float l = 56.5369;  
cout << "k=" << k << ".00 +-0.01 mm\n" << endl;  
cout << "k=" << l << k << "+-0.01mm" << endl << "zadanie2";
```

В консоли будет следующее:

k=5.00 +-0.01 mm

k=56.53695+-0.01mm

zadanie2