**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

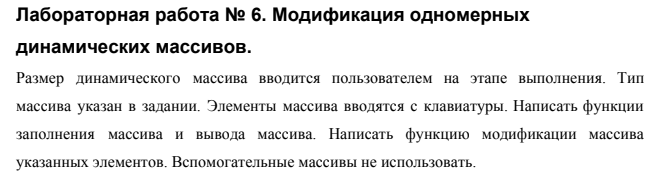
**По лабораторной работе № 6**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Индюченко Никита Андреевич | БПМ211 | 16.12.2021 |
|  |
|  |
|  |

**Москва – 2021 г.**

**ЗАДАНИЕ (вариант №12)**

****

*Текст задания Вашего варианта*

****

**РЕШЕНИЕ**

Код программы с комментариями

#include <stdio.h>

#include <cstdlib>

void create\_arr(double\* arr, int \_size\_arr);

void print\_arr(double\* arr, int \_size\_arr);

int first\_negative\_arr(double\* arr, int \_size\_arr);

int main()

{

int size\_arr = 0;// кол-во элементов в массиве

printf("Enter size mass=");

scanf\_s("%d", &size\_arr); // ввод в консоль

double\* arr = (double\*)malloc(size\_arr\*sizeof(double)); // объявление динам. массива с использованием sizeof, чтобы узнать размер double, также malloc для выделение памяти

if (arr == NULL) // проверка на выделение памяти

{

printf("Memory no allocated\n");

return 1;

}

create\_arr(arr, size\_arr);// вызов функции для заполнения массива, где arr - указатель на 1-ый элемент массива, а size\_arr - кол-во элементов

printf("\n");

size\_arr=first\_negative\_arr(arr, size\_arr);// присвоение size\_arr размер модифицированного массива

print\_arr(arr, size\_arr);// вывод массива с новым размером

free(arr); //освобождение памяти

}

void create\_arr(double\* arr, int \_size\_arr)

{

for (int i = 0; i < \_size\_arr; i++)

{

double tmp;

printf("mass[%d]=",i);

scanf\_s("%lf", &tmp);

arr[i] = tmp;

}

}

void print\_arr(double\* arr, int \_size\_arr)

{

for (int i = 0; i < \_size\_arr; i++)

{

printf("mass[%d]=",i);

printf("%lf\n", arr[i]);

}

}

int first\_negative\_arr(double\* arr, int \_size\_arr)

{

int ind\_max = 0, max = arr[0]; // индекс максимального элемента и его значение равный по умолчанию нулевому элементу массива

for (int i = 0; i < \_size\_arr; i++)

{

if (max < arr[i])

{

max = arr[i];

ind\_max = i;

}

}

int size = \_size\_arr;// будет хранить размер обработанного массива

int count\_del = 0; // кол-во элементов, которые надо удалить

for (int i = 0; i < ind\_max; i++)

{

if (arr[i] < 0)

{

count\_del++;

}

}

size -= count\_del;

int now\_del = 0; // кол-во элементов, которые уже удалились

if (count\_del != 0) // если нашлись элементы, которые надо удалить

{

for (int i = 0; i < \_size\_arr; i++)

{

if (arr[i] < 0)

{

for (int j = i; j < \_size\_arr; j++)// сдвиг элементов массива влево

{

arr[j] = arr[j + 1];

}

now\_del++;

i--;

if (now\_del == count\_del) // если кол-во удалённых элементов = эл. сколько нужно удалить, то выходим из цикла

{

break;

}

}

}

for (int i = size; i < \_size\_arr; i++) // "мусор", который образовался при сдвиге заполняем 0

{

arr[i] = 0;

}

}

return size; // возвращение нового размера массива

}

**ТЕСТЫ**

**Тест № 1**

**Проверка на выделение памяти**

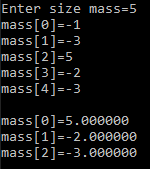
*Результаты теста 1*

**

**Тест № 2**

**Удаление отрицательных элементов, до первого максимального**

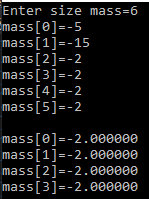
*Результаты теста 2*

**

**Тест № 3**

**Удаление отрицательных элементов, если массив состоит только из отрицательных чисел**

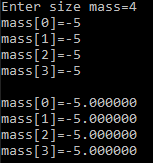
*Результаты теста 3*

**

**Тест № 4**

**Если все элементы отрицательные и одинаковые**

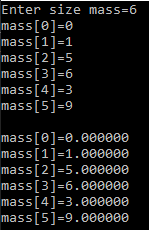
*Результаты теста 4*

**

**Тест № 5**

**Если все элементы >=0**

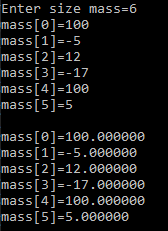
*Результаты теста 5*

****

**Тест № 6**

**Если первый элемент является максимальным**

*Результаты теста 6*

**

**Тест № …**

*Результаты теста …*