**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнила:

Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 14

1)В массиве, состоящем из вещественных элементов, вычислить количество элементов массива, больших некоторого заданного числа **С**, и произведение элементов массива, расположенных после максимального по модулю элемента.

#include <Windows.h>

#include <stdio.h>

#include <stdlib.h>

int count\_elements(float\* arr, int size, float c) {//определение функции count\_elements, которая принимает массив чисел типа float, его размер и значение c, и возвращает количество элементов массива, больших чем c.

int count = 0;

for (int i = 0; i < size; i++) {

if (arr[i] > c) {

count++;

}

}

return count;

}

float multiply\_elements(float\* arr, int size) {// определение функции multiply\_elements, которая принимает массив чисел типа float и его размер, находит элемент с наибольшим значением, затем находит произведение всех элементов массива после этого элемента и возвращает его.

float max\_value = arr[0];

int max\_index = 0;

for (int i = 1; i < size; i++) {

if ((arr[i]) > (max\_value)) {

max\_value = arr[i];

max\_index = i;

}

}

float product = 1;

for (int i = max\_index + 1; i < size; i++) {

product \*= arr[i];

}

return product;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size;

float c;

printf("Введите длинну массива: ");

scanf\_s("%d", &size);

float\* arr = (float\*)malloc(size \* sizeof(float)); //выделяет блок памяти размером size \* sizeof(float) байт и возвращает указатель на начало этого блока, который затем приводится к типу float\* и сохраняется в переменной arr

printf("Введите элементы массива:\n");

for (int i = 0; i < size; i++) {

scanf\_s("%f", &arr[i]);

}

printf("значение C: ");

scanf\_s("%f", &c);

int count = count\_elements(arr, size, c);

float product = multiply\_elements(arr, size);

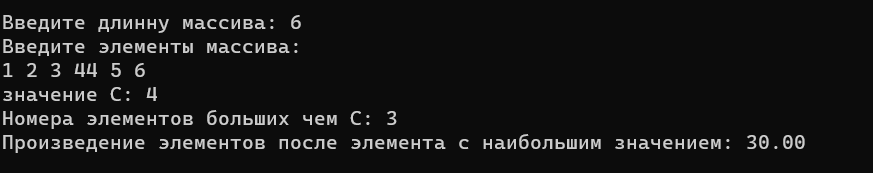
printf("Номера элементов больших чем C: %d\n", count);

printf("Произведение элементов после элемента с наибольшим значением: %.2f\n", product);

free(arr);//Освобождение выделенной памяти для массива с помощью функции free.

return 0;

}



2. Проверить, есть ли в матрице хотя бы одна строка, содержащая элемент, равный нулю, и найти ее номер. Уменьшить все элементы матрицы на значение первого элемента найденной строки

#include <iostream>

#include <iomanip>

#include <Windows.h>

#include <stdio.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, m;

cout << "Введите размер матрицы ";

cin >> n >> m;

int\*\* matrix = new int\* [n];//динамическое выделение памяти для двумерного массива matrix с помощью оператора new.

cout << "Введите значения элементов матрицы " << '\n';

for (int i = 0; i < n; i++) {

matrix[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> matrix[i][j];

}

}

bool t = false;// поиск элемента со значением 0 в матрице, и если такой элемент найден, то выводится номер строки, в которой он находится, и значение первого элемента этой строки.

int num\_str = 0;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (matrix[i][j] == 0) {

t = true;

num\_str = i;

break;

}

}

}

if (!t) {

cout << "В матрице нет элементов со значением 0" << '\n';

}

else {

cout << "Номер строки, в которой есть элемент со значением 0 = " << num\_str + 1 << '\n';

cout << "Значение первого элемента этой строки = " << matrix[num\_str][0] << '\n';

cout << "Новая матрица: " << '\n';

for (int i = 0; i < n; i++) {//вывод новой матрицы, где каждый элемент вычисляется как разница между его значением и значением первого элемента строки, в которой был найден 0.

for (int j = 0; j < m; j++) {

cout << matrix[i][j] - matrix[num\_str][0] << " ";

}

cout << '\n';

}

}

delete[] matrix;//освобождается выделенная память с помощью оператора delete[] matrix.

}

