**Отчет**

**по Лабораторной работе № 1**

**«Линейные программы»**

Специальность 09.02.03«Программирование в компьютерных системах»

ПМ.01 «Разработка программных модулей и программного обеспечения для компьютерных систем»

МДК.01.02 «Прикладное программирование»

**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

Выполнил:

студент группы 2337

Коряков С.А.

Санкт-Петербург

2018/2019

Преподаватель:

ФИО Сорокин Д. Ю.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_201\_\_г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Цель работы

Продемонстрировать работу с механизмами динамического выделения и освобождения памяти на примере решения задач с одномерными массивами.

Практическое задание

Составить программу, которая:

1. запрашивает у пользователя размерность массива чисел и динамически выделяет память для хранения указанного количества значений.

2. запрашивает у пользователя массив чисел и выводит этот массив в на экран. Среди чисел должны быть отрицательные, положительные и равные нулю. Дополнительные параметры, если они упоминаются в тексте задания, вводятся с клавиатуры.

3. вычисляет значения переменных согласно варианту задания.

4. выводит значения всех полученных переменных с пояснениями.

Индивидуальное задание

8. Даны вещественные массивы A[15], W[15]. Для каждого массива определить количество элементов, предшествующих наибольшему элементу массива и их среднее геометрическое.

Блок-схема алгоритма работы программы

Листинг

#include <iostream> // подключение основной библиотеки для работы основных функций

#include <math.h> // подключение библиотеки для математических вычислений

#include <limits> // подключение библиотеки для задания минимальных значений

using namespace std; // использование постоянного пространства имен для удобного пользования

void main() // главная область действия программы

{float \*A, \*W, maxa= numeric\_limits<float>::min(), maxw= numeric\_limits<float>::min(), geoma=1,geomw=1; //создание двух указателей для массивов, переменных для нахождения максимального значения и двух переменных для среднего геометрического

int n = 0,numba,numbw; // создание переменной, задающая длину массивов, и переменных хранящих позицию максимального элемента массивов

cout << "vvedite dlinu oboih massivov:";

cin >> n;

A = new float[n]; // создание массива А

W = new float[n]; // создание массива W

for (int i = 0; i < n; i++) // Цикл ввода массива А

{cout << "Please enter A[" << i << "] number "; // Сообщение о вводе i-того элемента

cin >> A[i]; // Ввод i-того элемента}

for (int i = 0; i < n; i++) // Цикл ввода массива W

{cout << "Please enter W[" << i << "] number "; // Сообщение о вводе i-того элемента

cin >> W[i]; // Ввод i-того элемента

}

for (int i = 0; i < n; i++) // цикл по поиску максимальных элементов с их адресами

{if (A[i] > maxa) //анализ переменной, по сравнению с предыдущей на то является ли она она больше

{maxa = A[i]; //запись наибольшее значение массива А на данный момент итерации

numba = i; // запись номера наибольшего значения массива А на данный момент итерации

}

if (W[i] > maxw) //анализ переменной, по сравнению с предыдущей на то является ли она она больше

{maxw = W[i]; //запись наибольшее значение массива W на данный момент итерации

numbw = i; // запись номера наибольшего значения массива W на данный момент итерации

}}

cout << "Kol-vo el. v massive A do max. elementa :"<<numba << endl; // вывод количества элементов до максимального в массива А

cout << "Kol-vo el. v massive W do max. elementa :" << numbw << endl; // вывод количества элементов до максимального в массива W

for (int i = 0; i < numba; i++) // цикл для перемножение элементов до максимального в массиве А

{geoma = A[i] \* geoma; //умножение элемента под номером i с общим произведением элементов

}

cout << "srednee geometricheskoe pervih " << numba << "elementov v Massive A = " << pow(geoma, float(1) / numba) << endl; //финальное вычисление и вывод среднего геометрического в массиве А

for (int i = 0; i < numbw; i++) // цикл для перемножение элементов до максимального в массиве W

{geomw = W[i] \* geomw; //умножение элемента под номером i с общим произведением элементов

}

cout << "srednee geometricheskoe pervih " << numbw << "elementov v Massive W = " << pow(geomw, float(1) / numbw) << endl; //финальное вычисление и вывод среднего геометрического в массиве W

delete [] A; //очищение памяти от массива А

delete[] W; //очищение памяти от массива W

}

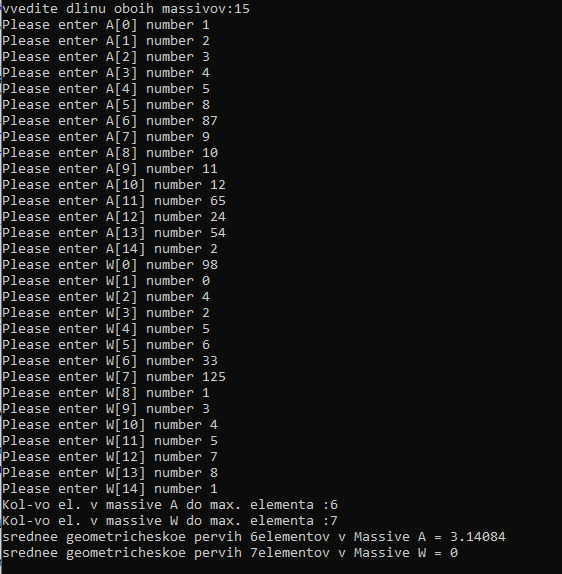
Протокол выполнения

Рисунок 2, протокол выполнения программы