Звіт з лабораторної роботи #8

з курсу “Програмування”

студента групы ПА-18-3

Дуброва Романа Олеговича

Кафедра компьютерных технлогий, ДНУ

2018/2019 у.г

**1. Постановка задачи(вариант 12)**

Складіть програму, яка здійснює обробку вхідних даних за допомогою одновимірних масивів. Для вводу даних і виводу результатів програма використовує текстові файли за допомогою перенапрямлення у командному рядку; ввід з клавіатури також можливий.

**Задайте два вектори А(n) і В(m). Виконайте такі завдання:**

1. визначити вектор С(n+m) із елементів, які у векторах А і В не повторюються. Елементи вектора С розташувати в порядку зростання;

2. замінити мінімальний елемент вектора А на середнє арифметичне елементів вектора В;

3. визначити, чи є послідовність елементів векторів А і В упорядкованою за убуванням.

**2. Описание решения**

Задание 1: Для начала мы создаем новый массив размера (n + m) и записываем в него все элементы массивов A и B. Далее сортируем этот массив по возрастанию и действуем по следующему алгоритму: мы смотрим на первый элемент массива и сравниваем его со следующим. Если они равны, то мы записываем значение первого элемента в переменную temp. Далее, мы смещаем весь массив на 1 элемент влево, а значение temp сохраняем в конец массива. Переменная count отвечает за количество элементов массивов до тех, которых мы поместили в конец. Ее мы уменьшаем на 1, т.к один элемент поместился в конец. Так же, мы присваиваем переменной trigger значение 1, отмечая этим, что следующий элемент после первого так же имеет копию. Далее, мы действуем по такому же алгоритму и получаем массив, где в левой части находится count элементов, которые не повторяются, а в правой части (n + m – count) элементов, которые повторяются. Далее, мы обнуляем все элементы в правой части, а левую часть выводим на экран.

Задание 2: Мы находим минимальный элемент в массиве А и присваиваем ему значение среднего арифметическое элементов массива B.

Задание 3: Мы создаем массив С и последовательно сохраняем в нем элементы массивов А и В. Далее, мы сравниваем каждый элемент полученного массива с последующего. Если он больше, последующего, то мы выходим из цикла и выводим соответственное сообщение на экран, иначе мы проходим через весь цикл и выводим на экран сообщение о том, что послед. Убывающая.

**3. Выходной текст программы**

#include "pch.h"

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

void printArray(float\* array, int size) {

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

cout << array[i] << " ";

}

cout << endl;

}

void swap(float\* a, float\* b) {

float temp = \*a;

\*a = \*b;

\*b = temp;

}

int findMinElement(float\* array, int size, int i) {

int min = i;

for (; i < size; i++) {

if (array[i] < array[min])

min = i;

}

return min;

}

void sort(float\* array, int size) {

int min;

for (int i = 0; i < size - 1; i++) {

min = findMinElement(array, size, i);

swap(array[min], array[i]);

}

}

float arrayAverage(float\* array, int size) {

float sum = 0;

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

sum = sum + array[i];

}

return sum / size;

}

void firstTask(float\* firstArray, float\* secondArray, int n, int m) {

float\* resultArray = new float[n + m];

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

resultArray[i] = firstArray[i];

}

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

resultArray[i] = secondArray[i - n];

}

sort(resultArray, m + n);

int count = n + m;

int trigger = 0;

int temp;

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

if (resultArray[i] == resultArray[i + 1]) {

temp = resultArray[i];

for (int j = i; j < count; j++) {

resultArray[j] = resultArray[j + 1];

}

count--;

i--;

resultArray[count] = temp;

trigger = 1;

}

else if (trigger == 1) {

temp = resultArray[i];

for (int j = i; j < count; j++) {

resultArray[j] = resultArray[j + 1];

}

count--;

i--;

resultArray[count] = temp;

trigger = 0;

}

}

for (int i = count; i < m + n; i++) {

resultArray[i] = 0;

}

if (count == 0) {

cout << "В массиве С нет элементов.\n\n";

}

else {

cout << "Массив С: ";

printArray(resultArray, count);

cout << endl;

}

delete[] resultArray;

}

void secondTask(float\* firstArray, float\* secondArray, int n, int m) {

int min = findMinElement(firstArray, n, 0);

firstArray[min] = arrayAverage(secondArray, m);

}

void thirdTask(float\* firstArray, float\* secondArray, int n, int m) {

float\* resultArray = new float[m + n];

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

resultArray[i] = firstArray[i];

}

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

resultArray[i] = secondArray[i - n];

}

bool trigger = 1;

for (int i = 0; i < m + n - 1; i++) {

if (resultArray[i] < resultArray[i + 1]) {

trigger = 0;

break;

}

}

cout << "Массив C: ";

printArray(resultArray, n + m);

trigger ? cout << "Массив С образует убывающую последовательность.\n\n" : cout << "Массив С НЕ образует убывающую последовательность\n\n";

delete[] resultArray;

}

int main() {

setlocale(LC\_ALL, "Russian");

int n, m;

cout << "Если вы не использовали текстовый файл для ввода, то введите нужные данные вручную.\n";

cout << "Вам нужно ввести два массива А и B размером N и M, который вам так же следует ввести.\n";

cout << "Выберите путь ввода данных.\n1.Ввод данных из текстового файла.\n2.Ввод данных с консоли.\nВаш выбора: ";

int c;

cin >> c;

if (c == 1) {

fstream file;

file.open("input.txt");

file >> n;

float\* firstArray = new float[n];

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

// firstArray[i] = new float[n];

//}

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

file >> firstArray[i];

}

file >> m;

float\* secondArray = new float[m];

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

// file >> secondArray[i];

//}

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

file >> secondArray[i];

}

system("cls");

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;

cout << "Задание 1: \n";

firstTask(firstArray, secondArray, n, m);

cout << "Задание 2: \n";

secondTask(firstArray, secondArray, n, m);

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;

cout << "Задание 3: \n";

thirdTask(firstArray, secondArray, n, m);

delete[] firstArray;

delete[] secondArray;

file.close();

}

else if (c == 2) {

cout << "Кол-во элементов в массиве A: ";

cin >> n;

float\* firstArray = new float[n];

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

cout << "A[" << i << "] = ";

cin >> firstArray[i];

}

cout << "Кол-во элементов в массиве B: ";

cin >> m;

float\* secondArray = new float[m];

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

cout << "B[" << i << "] = ";

cin >> secondArray[i];

}

system("cls");

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;

cout << "Задание 1: \n";

firstTask(firstArray, secondArray, n, m);

cout << "Задание 2: \n";

secondTask(firstArray, secondArray, n, m);

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;

cout << "Задание 3: \n";

thirdTask(firstArray, secondArray, n, m);

delete[] firstArray;

delete[] secondArray;

}

/\*cout << "Кол-во элементов в массиве A: ";

cin >> n;

float\* firstArray = new float[n];

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

cout << "A[" << i << "] = ";

cin >> firstArray[i];

}

cout << "Кол-во элементов в массиве B: ";

cin >> m;

float\* secondArray = new float[m];

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

cout << "B[" << i << "] = ";

cin >> secondArray[i];

}\*/

/\*system("cls");

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;\*/

/\*cout << "Задание 1: \n";

firstTask(firstArray, secondArray, n, m);

cout << "Задание 2: \n";

secondTask(firstArray, secondArray, n, m);

cout << "Массив A: ";

printArray(firstArray, n);

cout << "Массив B: ";

printArray(secondArray, m);

cout << endl;

cout << "Задание 3: \n";

thirdTask(firstArray, secondArray, n, m);

delete[] firstArray;

delete[] secondArray;\*/

return 0;

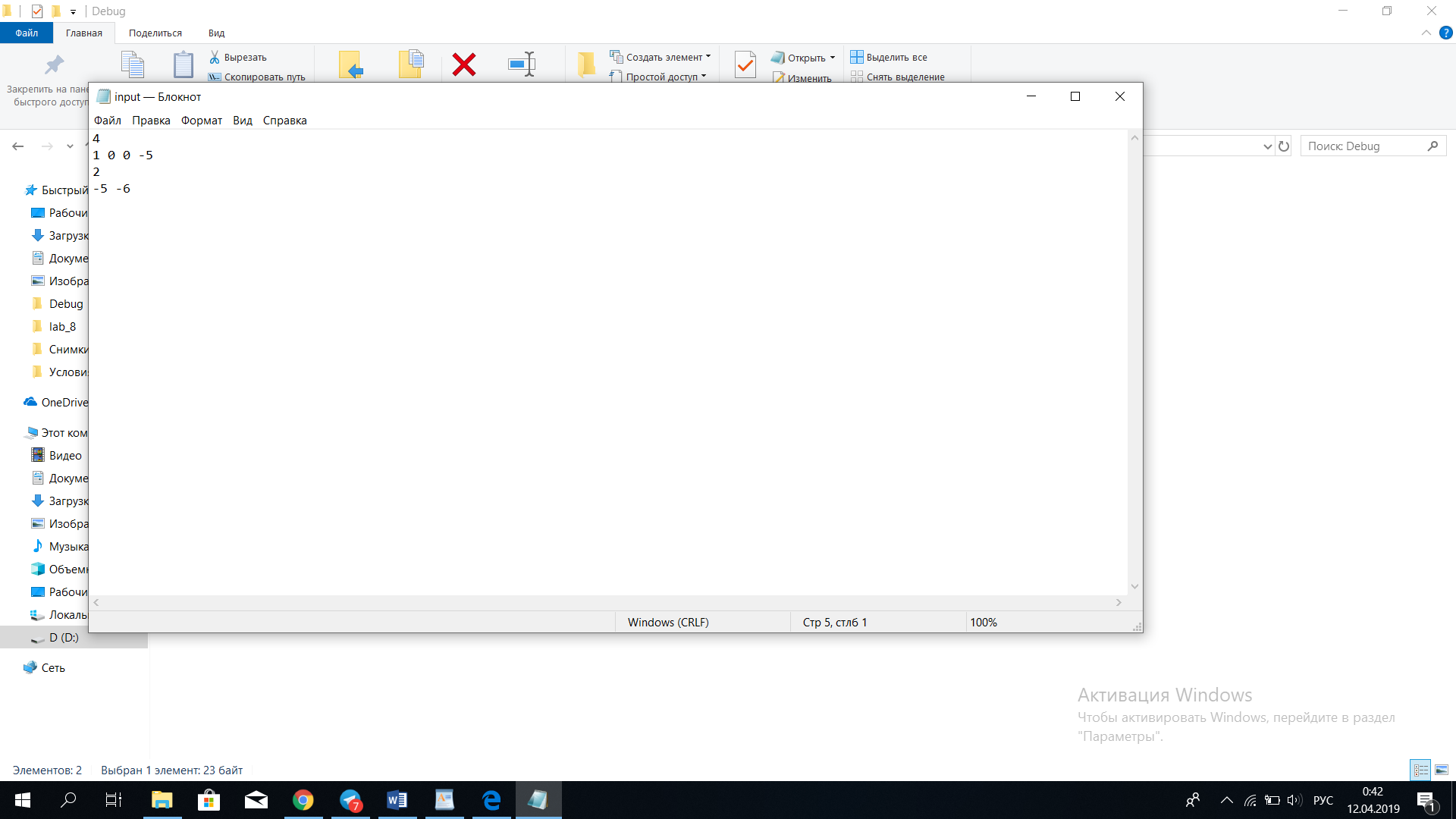
}

**4. Описание интерфейса**

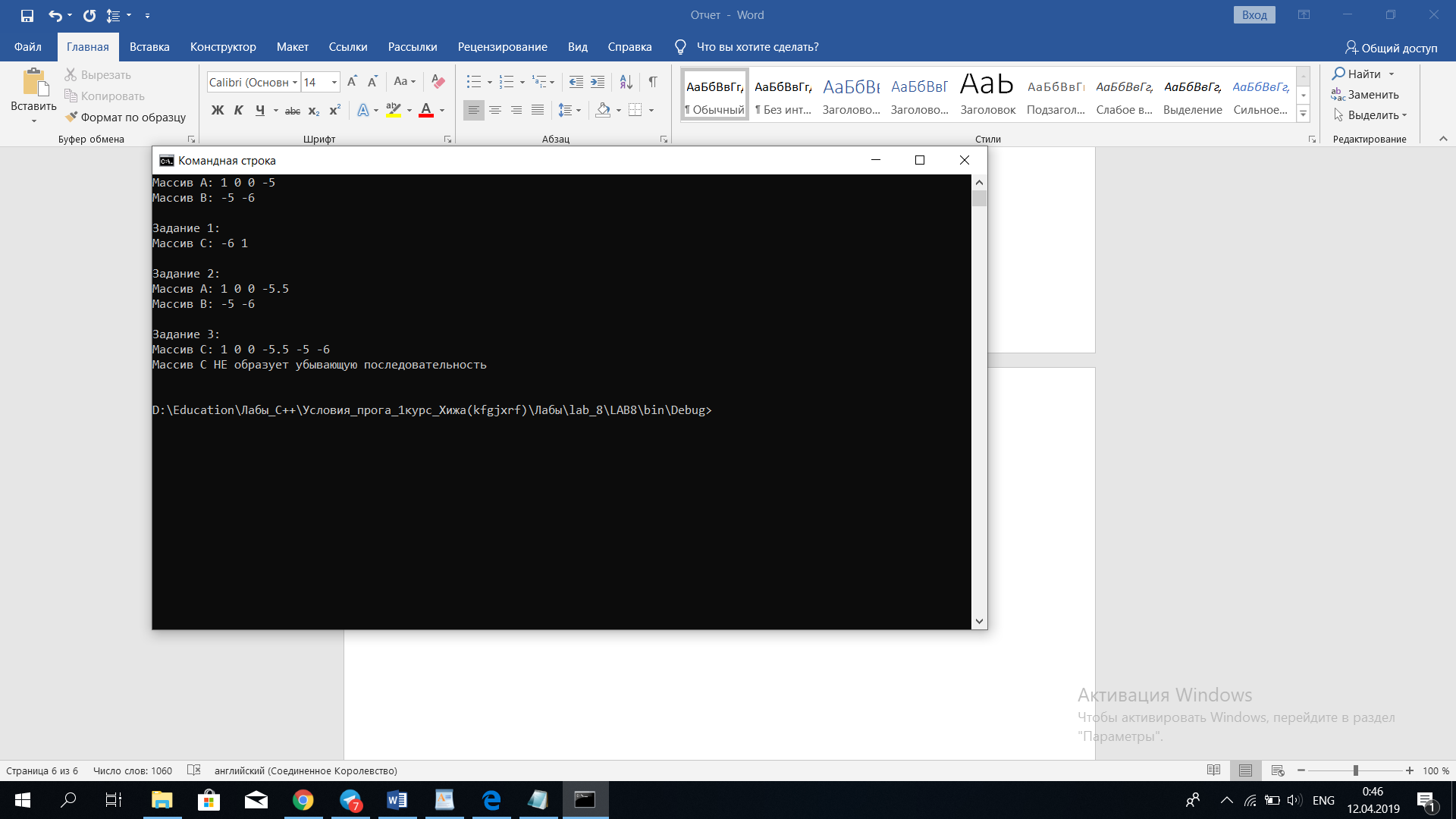
Пользователь должен ввести размеры двух массивов и их элементы. Он может это сделать, перенаправив поток ввода с клавиатуры на файл. В таком случае, на экран просто выведется результат выполнения программы. Иначе, пользователю будет предложено ввести самостоятельно данные и результат работы программы так же будет выведен на экран.

**5. Описание тестовых примеров**

Тест ввода данных с файла.

Данные в файле input.txt

Результат работы программы со входными данными из input.txt



Тест ввода данных вручную:

