**Лабораторная работа №7**

**по курсу**

**«Введение в программирование»**

**Вариант 8**

**Выполнил:**

Студент группы ЭР-05-20

Волчков Дмитрий

**Задание:**

*Общее задание:*

Даны две последовательности. Изменить первую последовательность, вставляя после каждого элемента 1-ой последовательности очередной элемент 2-ой   последовательности, например

*1 2 3 4 -5*

*6 8 10 -20 33*

Результат:

*1 6 2 8 3 10 4 -20 -5 33*

*Индивидуальное задание:*

Найти максимальный отрицательный элемент в массиве с нечетным индексом. Также найти среднее арифметическое таких элементов, начиная от индекса минимального элемента массива.

1. **Постановка задачи:**
   1. Разработать программу, которая считывает две последовательностях и вставляет поочередно элементы первой и второй последовательностей. В случаях, когда размер последовательности некорректен, просить изменить ошибку.

*Входные данные:* количество элементов исходных последовательностей (n типа int), первая последовательность (first типа vector<double>)

*Выходные данные:* конечная последовательность (result типа vector<double>)

*Ограничения:* количество элементов последовательностей не может быть меньше или равно 1. Длинны последовательностей равны и задаются пользователем вручную.

* 1. Разработать программу, которая считывает последовательность и ищет в ней максимальный отрицательный элемент в массиве, считая, что индекс первого элемента последовательности равен нулю, и среднее арифметическое отрицательных элементов, начиная с индекса минимального элемента массива.

Входные данные: количество элементов последовательности (n типа int) и последовательность (third типа vector<double>)

Выходные данные: максимальный отрицательный элемент с нечетным индексом (maxmin типа double), минимальный элемент, индекс минимального элемента (min\_ind типа int) и среднее арифметическое отрицательных элементов начиная от индекса минимального элемента массива (sr\_ar типа double)

*Ограничения:* количество элементов последовательности задаются пользователем, количество элементов последовательности не могут быть меньше или равны 1.

Вид приложения – консольное приложение на языке C++

Среда разработки – Microsoft Visual Studio 2019

1. **Разработка программы:**

Для решения первой части задания было решено взять всего два вектора: первый и конечный, вторая последовательность вводится напрямую в конечный вектор. Было решено сделать проверку на ввод количества элементов последовательности.

Для решения второй части задания было решено взять один вектор, необходимости в нескольких нет. Было решено сделать проверку на ввод количества элементов последовательности. Также, если в программе нет отрицательных элементов или нет отрицательных элементов с нечетным индексом, то вывести что отрицательных элементов нет или нет отрицательных элементов с нечетным индексом соответственно.

1. **Состав данных:**

*Для первой программы (первой части задания):*

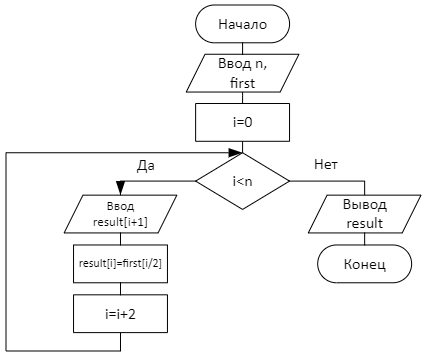
|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| first | первая последовательность | вещественный | вектор STL |
| n | количество элементов последовательности | целый | простая переменная |
| Промежуточные данные | | | |
| i | счетчик цикла | целый | простая переменная |
| Выходные данные | | | |
| result | конечная последовательность | вещественный | вектор STL |

*Для второй программы (второй части задания):*

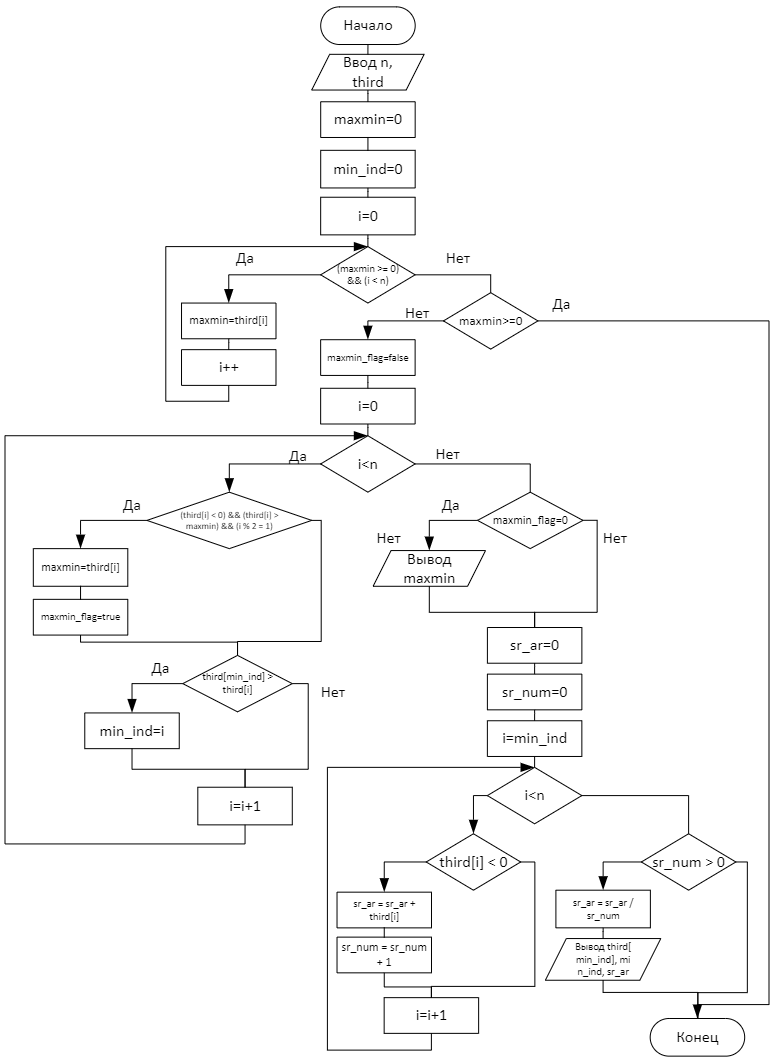
|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| Исходные данные | | | |
| third | первая последовательность | вещественный | вектор STL |
| n | количество элементов последовательности | целый | простая переменная |
| Промежуточные данные | | | |
| i | счетчик цикла | целый | простая переменная |
| maxmin\_flag | флаг нахождения максимального отрицательного числа | логический | простая переменная |
| sr\_num | количество отрицательных элементов с индекса минимального элемента | целый | простая переменная |
| Выходные данные | | | |
| maxmin | максимальный отрицательный элемент с нечетным индексом | вещественный | простая переменная |
| min\_ind | индекс минимального элемента | целый | простая переменная |
| sr\_ar | среднее арифметическое отрицательных элементов с индекса минимального элемента | вещественный | простая переменная |

1. **Алгоритм программы:**

*Алгоритм первой программы (первой части задания):*



*Алгоритм второй программы (второй части задания):*



1. **Листинг программы:**

*Листинг первой программы (первой части задания):*

#include <iostream>

#include <vector>

/\*Общая часть задания :

Даны две последовательности. Изменить первую последовательность, вставляя после каждого элемента 1-ой

последовательности очередной элемент 2-ой последовательности.\*/

int main()

{

setlocale(0, "");

int n;

std::cout << "Введите размер последовательностей:\n";

std::cin >> n;

while (n <= 1) { // Проверка на некорректный размер последовательности

std::cout << "Введите корректный размер последовательности:\n";

std::cin >> n;

}

std::vector <double> first(n), result(n\*2);

std::cout << "Введите элементы первой последовательности\n";

for (int i = 0; i < n; i += 1) std::cin >> first[i];

std::cout << "Введите элементы второй последовательности\n";

for (int i = 0; i < n \* 2; i += 2) {

std::cin >> result[i + 1];

result[i] = first[i / 2];

}

std::cout << "Полученная последовательность:\n";

for (int i = 0; i < n \* 2; i += 1) std::cout << result[i] << " ";

std::cout << std::endl;

system("pause");

return 0;

}

*Листинг второй программы (второй части задания):*

#include <iostream>

#include <vector>

#include <iomanip>

/\* 8. Найти максимальный отрицательный элемент в массиве с нечетным индексом.

Также найти среднее арифметическое таких элементов, начиная от индекса минимального элемента массива.\*/

int main()

{

setlocale(0, "");

std::cout << "Введите количество элементов последовательности:\n";

int n;

std::cin >> n;

while (n <= 1) { // Проверка на некорректный размер последовательности

std::cout << "Введите корректный размер последовательности:\n";

std::cin >> n;

}

std::vector <double> third(n);

std::cout << "Введите последовательность:\n";

for (int i = 0; i < n; i += 1) std::cin >> third[i];

double maxmin = 0;

int min\_ind = 0;

int i = 0;

// Поиск первого отрицательного элемента

while ((maxmin >= 0) && (i < n)) {

maxmin = third[i];

i++;

}

if (maxmin >= 0) std::cout << "Отрицательных элементов в последовательности нет\n";

else {

// Поиск максимального отрицательного элемента и индекса минимального элемента

bool maxmin\_flag = false;

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

if ((third[i] < 0) && (third[i] > maxmin) && (i % 2 == 1)) {

maxmin = third[i];

maxmin\_flag = true;

}

if (third[min\_ind] > third[i]) min\_ind = i;

}

if (maxmin\_flag) std::cout << "Максимальный отрицательный элемент с нечетным индексом: " << maxmin << std::endl;

else std::cout << "Отрицательный элемент с нечетным индексом отсутствует\n";

// Нахождение среднего арифметического отрицательных элементов начиная от индекса минимального элемента массива

double sr\_ar = 0;

int sr\_num = 0;

for (int i = min\_ind; i < n; i += 1) if (third[i] < 0) {

sr\_ar = sr\_ar + third[i];

sr\_num = sr\_num + 1;

}

if (sr\_num > 0) {

sr\_ar = sr\_ar / sr\_num;

std::cout << "Минимальный элемент: " << third[min\_ind] << "\t С индексом: " << min\_ind << std::endl;

std::cout << "Среднее арифметическое отрицательных элементов начиная от индекса минимального элемента массива: " << std::setprecision(4) << sr\_ar << std::endl;

}

else std::cout << "Таких элементов нет\n";

}

system("pause");

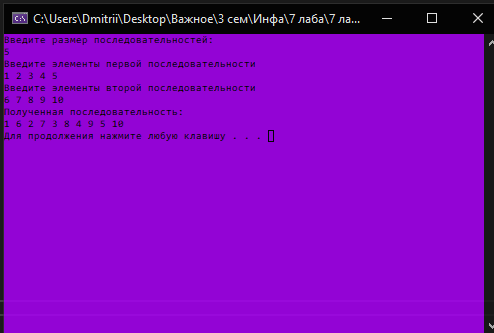
return 0;

}

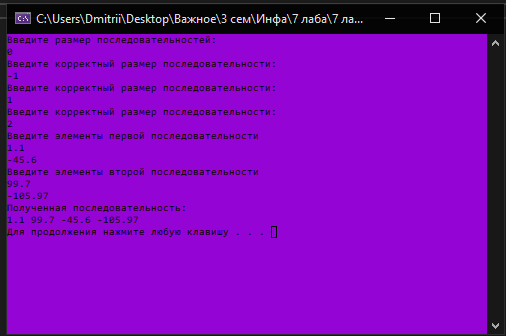
1. **Контрольные тесты**

*Контрольные тесты для первой программы (первой части задания):*

Тест №1:

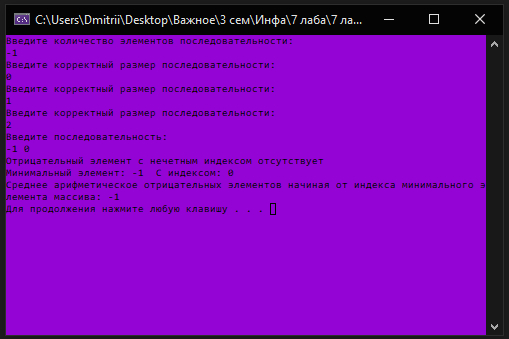


Тест №2:

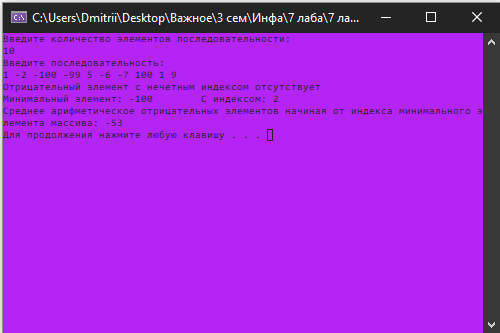


*Контрольные тесты для второй программы (второй части задания):*

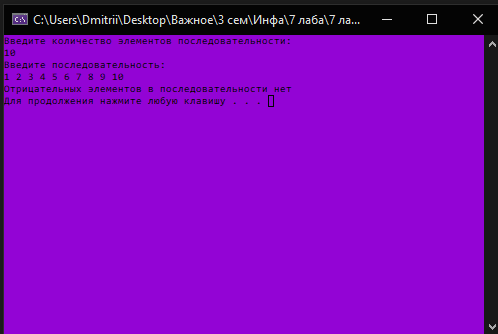
Тест №1:



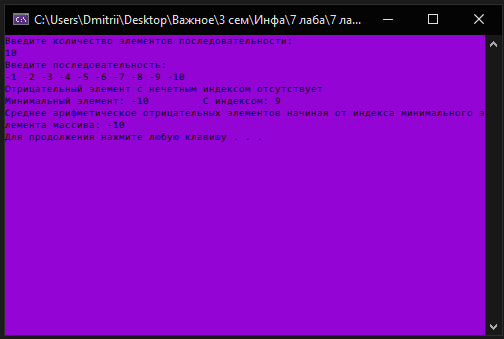
Тест №2:



Тест №3:



Тест №4:



Тест №5:

