ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка числовых последовательностей |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Н.А.Костяков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

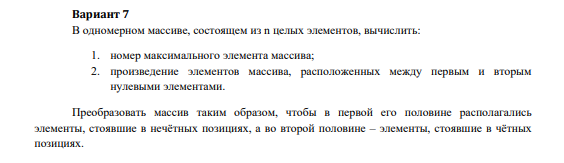
Санкт-Петербург 2022

**Цель работы**

Целью работы является изучение структуры данных одномерный массив.

**Задание на лабораторную работу**

Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамичес



**Ход выполнения**

**Используемые функции**

**Имя** fill()

**Назначение:** ввод с проверкой

**Входные данные :** запросы с клавиатуры

**Выходные данные:** число в double

**Побочные эффекты отсутствуют**

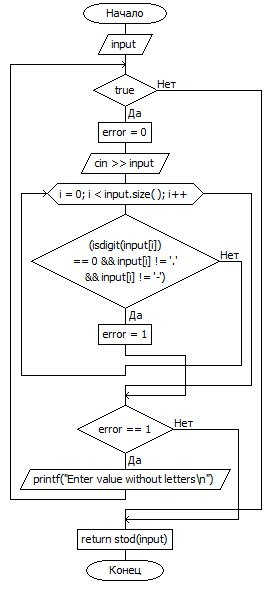
**Псевдокод**

Пока не введут значение без букв

Запрос ввода с клавиатуры

Вывод double

**Блок-схема**

****

**Имя** max\_i()

**Назначение:** Поиск индекса максимального элемента

**Входные данные :** массив и его длинна

**Выходные данные:** индекс максимального элемента

**Побочные эффекты отсутствуют**

**Псевдокод**

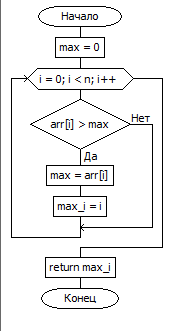
Для каждого элемента в массиве

Сравнить с переменной max

Если max меньше, перезаписать значение max и запомнить индекс

Вывод индекса

**Блок-схема**



**Имя** nul()

**Назначение:** Поиск произведения между нулями

**Входные данные :** массив и его длинна

**Выходные данные:** произведение чисел между нулями

**Побочные эффекты отсутствуют**

**Псевдокод**

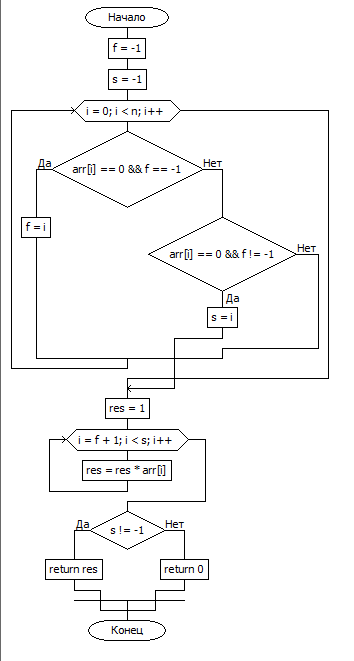
Поиск индексов нулей

Если найдено два нуля

Пройтись по всем индексам между нулями и перемножить

Вывести результат перемножения

**Блок-схема**



**Имя** mut()

**Назначение:** перестановка местами элементов массива

**Входные данные :** массив и его длинна

**Выходные данные:** новый массив

**Побочные эффекты отсутствуют**

**Псевдокод**

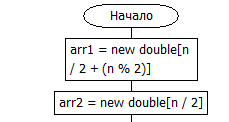
Создать три массива

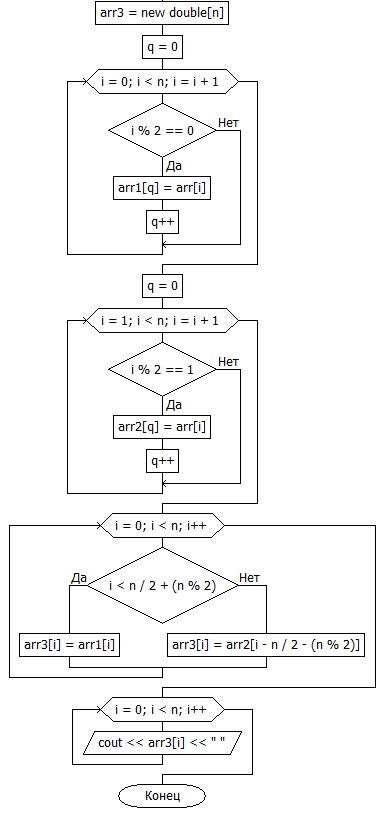
Наполнить два частями шаблона

Наполнить третий первым и вторым массивом

Вывести результат

**Блок-схема**





**Листинг всей программы**

#define \_CRTDBG\_MAP\_ALLOC

#include <stdlib.h>

#include <crtdbg.h>

#ifdef \_DEBUG

#ifndef DBG\_NEW

#define DBG\_NEW new ( \_NORMAL\_BLOCK , \_\_FILE\_\_ , \_\_LINE\_\_ )

#define newDBG\_NEW

#endif

#endif

#include <iostream>

#include <string>

using namespace std;

int max\_i(double arr[], int n) { //поиск индекса максимального элемента

double max = 0;

int max\_i;

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

if (arr[i] > max) {

max = arr[i];

max\_i = i;

}

}

return max\_i;

}

double nul(double arr[], int n) { //поиск произведения между нулями

int f = -1;

int s = -1;

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

if (arr[i] == 0 && f==-1) {

f = i;

}

else if (arr[i] == 0 && f != -1) {

s = i;

break;

}

}

double res = 1;

for (int i = f+1; i < s; i++) {

res = res \* arr[i];

}

if (s != -1) {

return res;

}

else {

return 0;

}

}

void mut(double arr[],const int n) {//переставление элементов

double\* arr1 = new double[n / 2 +(n%2)];

double\* arr2 = new double[n / 2];

double\* arr3 = new double[n];

int q = 0;

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

if (i % 2 == 0) {

arr1[q] = arr[i];

q++;

}

}

q = 0;

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

if (i % 2 == 1) {

arr2[q] = arr[i];

q++;

}

}

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

if (i < n / 2 + (n % 2)) {

arr3[i] = arr1[i];

}

else

{

arr3[i] = arr2[i - n / 2 - (n % 2)];

}

}

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

cout << arr3[i]<<" ";

}

delete [] arr1;

delete [] arr2;

delete [] arr3;

}

double fill() {

std::string input;

while (true) //проверка введенного числа

{

bool error = 0;

std::cin >> input;

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

if ((isdigit(input[i]) == 0 && input[i] != '.' && input[i] != '-')) {

error = 1;

break;

}

}

if (error == 1) {

printf("Enter value without letters\n");

}

else

{

break;

}

}

return stod(input);

}

int main()

{

double n;

n = 0;

while(n==0 || n<0 ){

std::cout << "Enter N\n";

n = fill();

}

std::cout << "Enter arr 1 by 1\n";

double \*arr = new double[n]; //динамическое создание массива

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

arr[i]=fill();

}

cout << max\_i(arr, n)<<" Max elem" << endl; //результат

cout << nul(arr, n)<<"null multipl" << endl;

cout << "array mixing ";

mut(arr, n);

delete [] arr; //очистка

// Для обнаружения утечек памяти

\_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDOUT);

\_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);

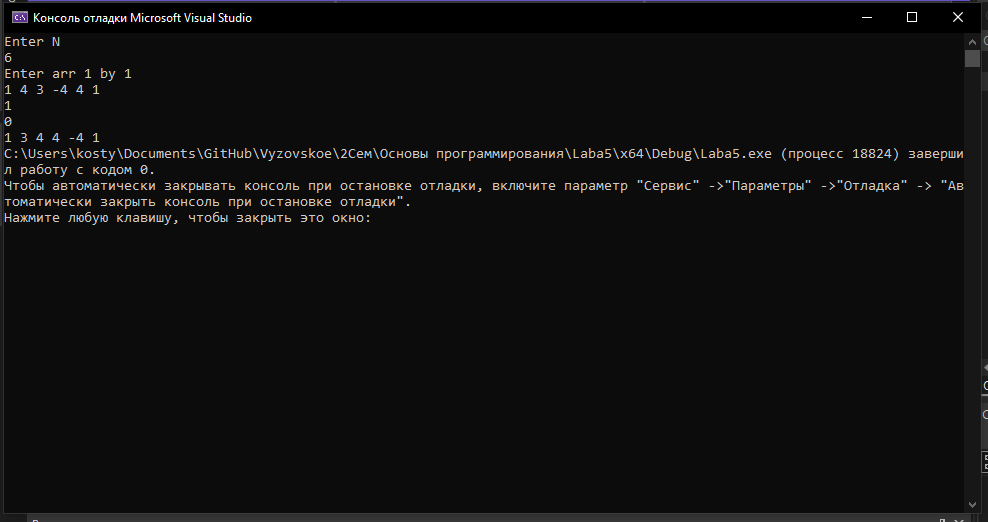
\_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDOUT);

\_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDOUT);

\_CrtDumpMemoryLeaks();

}**Результат программы**



**Вывод**

**Я изучил такую структуру данных как одномерный массив.**

Из достоинств программы можно выделить

- высокая скорость

- проверка введенных пользователем данных

Минусов нет