Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

Лабораторная работа №2. Алгоритмы поиска значений

тема

Преподаватель \_\_\_\_\_\_\_\_\_ Р.Ю. Царёв

подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031830504 \_\_\_\_\_\_\_\_\_ Е.В. Железкин

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2020

Содержание

[Содержание 2](#_Toc33628970)

[1 Цель работы 3](#_Toc33628971)

[2 Задача работы 3](#_Toc33628972)

[3 Ход работы 3](#_Toc33628973)

[4 Вывод 9](#_Toc33628974)

# Цель работы

Изучение алгоритмов поиска значений в массиве.

# Задача работы

Реализовать в программе два алгоритма (по выбору студента) из указанных ниже:

а) алгоритм поиска минимального (или максимального) значения,

б) алгоритм поиска моды,

в) алгоритм поиска медианы,

г) алгоритм поиска среднего значения,

д) алгоритм поиска значения, равного заданному.

Сравнить эффективность реализованных алгоритмов по времени их выполнения.

# Ход работы

Листинг 1 – содержание файла main.cpp проекта Lab\_2:

#include **<iostream>**#include **<string>**#include **<queue>**#include **<ctime>**#include **"funcs.h"**#define **LOCAL\_ERROR** -322  
#define **ENDLESS\_LOOP\_CONDITION** 0  
#define **ANOTHER\_VALUE** 54  
**using namespace** std;  
  
**int** main() {  
  
 **int** size = **LOCAL\_ERROR**;  
 string str;  
  
 cout << **"Введите размер массива: "**;  
 **while**(size == **LOCAL\_ERROR**)  
 {  
 getline(cin, str);  
 size = Checker(str);  
 **if** (size == **LOCAL\_ERROR**) cout << **"Некорректный ввод! Повторите попытку.\n"**;  
 }  
  
 **int** \*array = **new int**[size];  
  
 **int** temp = **LOCAL\_ERROR**;  
  
 cout << **"Введите элементы массива:\n"**;  
  
 **for** (**int** i = 0; i < size; i++){  
  
 temp = **LOCAL\_ERROR**;  
  
 **while** (temp == **LOCAL\_ERROR**){  
 getline(cin, str);  
 temp = Checker(str);  
 **if** (temp == **LOCAL\_ERROR**) cout << **"Некорректный ввод! Повторите попытку.\n"**;  
 }  
  
 array[i] = temp;  
  
 }  
  
 **int** loop = 0;  
 **char** let = 0;  
 queue<**int**> ans;  
 **int** time\_val = 0;  
  
 **while**(loop == **ENDLESS\_LOOP\_CONDITION**)  
 {  
 temp = **LOCAL\_ERROR**;  
 cout << **"Выберите действие: \n1 - Вычисление максимального/минимальн"  
 "ого элемента массива\n2 - Вычисление моды массива\n0 - Выход\n"**;  
 **while** (temp == **LOCAL\_ERROR**){  
 cin >> str;  
 temp = Checker(str);  
 **if** (temp == **LOCAL\_ERROR**) cout << **"Некорректный ввод! Повторите попытку.\n"**;  
 }  
  
 time\_val = time(0);  
  
 **if** (temp == 1) {  
 cout << **"Введите \"-\" для поиска минимального элемента массива/"  
 " \"+\" - максимального\n"**;  
  
 temp = **LOCAL\_ERROR**;  
  
 **while**(temp == **LOCAL\_ERROR**)  
 {  
 let = getchar();  
 **if** (let != **'+'** && let != **'-'**) cout << **"Некорректный ввод! Повторите попытку.\n"**;  
 **else** temp = ANOTHER\_VALUE;  
 }  
  
 **if** (let == **'+'**) cout << **"Максимальный элемент массива - "** << MMemSearch(array, size, **false**) << endl;  
 **if** (let == **'-'**) cout << **"Минимальный элемент массива - "** << MMemSearch(array, size, **true**) << endl;  
 cout << **"Время выполнения алгоритма - "** << time(0) - time\_val << endl << endl;  
  
 }  
  
 **if** (temp == 2){  
 ans = ModSearch(array, size);  
  
 cout << **"Моды данного массива: "**;  
  
 **while**(!ans.empty()){  
 cout << ans.front() << **"; "**;  
 ans.pop();  
 }  
 cout << endl;  
 cout << **"Время выполнения алгоритма - "** << time(0) - time\_val << endl << endl;  
 }  
  
 **if** (temp == 0) **return** 0;  
  
 }  
  
 **delete**[] array;  
 **return** 0;  
}

Листинг 2 – содержание файла funcs.cpp проекта Lab\_2:

#include **"funcs.h"**#include **<queue>**#include **<map>**#include **<string>  
  
int** MMemSearch(**int**\* array, **int** size, **bool** flag){  
  
 **int** result = 0;  
  
 **if** (flag == 0){  
 result = **MINIMAL\_BORDER**;  
 **for** (**int** i = 0; i < size; i++)  
 **if** (result < array[i]) result = array[i];  
 } **else** {  
 result = **MAXIMAL\_BORDER**;  
 **for** (**int** i = 0; i < size; i++)  
 **if** (result > array[i]) result = array[i];  
 }  
  
 **return** result;  
}  
  
std::queue<**int**> ModSearch(**int**\* array, **int** size){  
  
 std::map <**int**, **int**> temp;  
 **int** maxCount = 0;  
 std::queue<**int**> ret;  
  
 **for** (**int** i = 0; i < size; i++) {  
 temp[array[i]]++;  
 **if** (maxCount < temp[array[i]]) maxCount = temp[array[i]];  
 }  
  
 **for** (**auto** & i : temp)  
 **if** (i.second == maxCount) ret.push(i.first);  
  
 **return** ret;  
}  
  
**int** Checker(std::string tempForCheck)  
{  
 **int** number = 0;  
 **int** flag = 0;  
  
 **for** (**unsigned int** i = 0; i < tempForCheck.length(); i++)  
 {  
 **if** (i == 0 && tempForCheck[0] == **'-'**)  
 {  
 flag++;  
 **continue**;  
 }  
  
 **if** (tempForCheck[i] == **'0'** || tempForCheck[i] == **'1'** || tempForCheck[i] == **'2'** || tempForCheck[i] == **'3'** ||  
 tempForCheck[i] == **'4'** || tempForCheck[i] == **'5'** || tempForCheck[i] == **'6'** ||  
 tempForCheck[i] == **'7'** || tempForCheck[i] == **'8'** || tempForCheck[i] == **'9'**) number = number \* **ORDER** +  
 (tempForCheck[i] - **ASCII\_CONST**);  
 **else  
 return NUMBER\_ERROR**;  
 }  
  
 **if** (flag > 0) number \*= **NEGATIVE**;  
  
 **return** number;  
}

Листинг 3 – содержание файла funcs.h проекта Lab\_2:

#ifndef **LAB\_2\_FUNCS\_H**#define **LAB\_2\_FUNCS\_H**#define **MINIMAL\_BORDER** -9999999  
  
#define **NUMBER\_ERROR** -322  
  
#define **MAXIMAL\_BORDER** 9999999  
  
#define **ORDER** 10  
  
#define **NEGATIVE** -1  
  
#define **ASCII\_CONST** 48  
  
#include **<queue>**#include **<string>  
  
int** MMemSearch(**int**\* array, **int** size, **bool** flag);  
std::queue<**int**> ModSearch(**int**\* array, **int** size);  
**int** Checker(std::string tempForCheck);  
  
#endif *//LAB\_2\_FUNCS\_H*

1. Примеры работы:

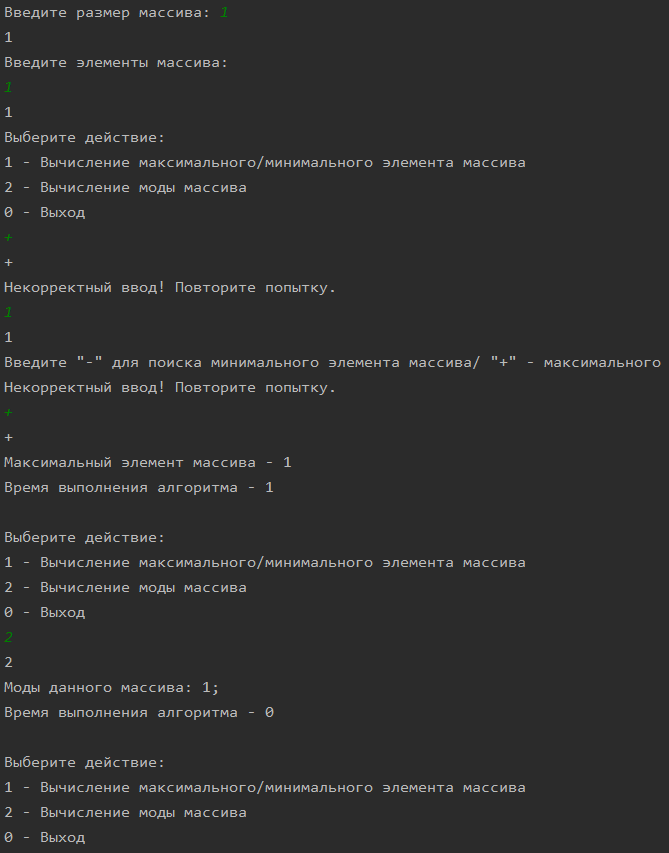


Рисунок 1 – результат выполнения запросов к массиву

# Вывод

В ходе данной лабораторной работы были реализованы алгоритмы поиска значений массива, описанные в задании.