Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Алгоритмы поиска значений
Тема

Преподаватель _____ Р. Ю. Царев
подпись, дата | инициалы, фамилия

Студент КИ19-17/1б 031939175 _____ А. Д. Непомнящий
номер группы, зачетной книжки | подпись, дата | инициалы, фамилия

Красноярск 2020

1 Цель работы

Изучение некоторых алгоритмов обработки массивов, нахождения характеристик массива, таких как медиана, среднее значение, мода, минимальный и максимальный элементы.

2 Задачи

Написать программу, реализующую алгоритм поиска медианы массива и среднего значения.

Предъявлены следующие требования к выполнению работы.

1. Строгое соответствие программы и результатов ее работы с полученным заданием.

2. Самостоятельные тестирование и отладка программы.

3. Устойчивость работы программы при любых воздействиях, задаваемых пользователем через интерфейс программы.

4. Предоставление демонстрационного примера и исходного текста программы для защиты.

5. Предоставление отчета по практическому заданию, содержащего описание реализованного алгоритма, программы, результатов работы программы (отчет необходимо загрузить на сайт курса).

3 Описание реализованного алгоритма

Реализованы алгоритмы нахождения необходимых значений. Поиск медиану выполняется упорядочением элементов массива по неубыванию и взятием среднего элемента. Если в выборке чётное число элементов, за медиану принимается полусумма двух соседних значений.

4 Описание программы

Для решения задачи была написана программа на языке C. Ниже приведен листинг кода.

Листинг 1 – Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>

enum Constants
{
    INPUT_SIZE = 100
};

int DefaultComp(const int* i, const int* j)
{
    return *i - *j;
}

int CycleInputInt(char* stringToOutput, bool(* pChecker)(int))
{
    int number;
    int position;
    char input[INPUT_SIZE];

    while (true)
    {
        printf("%s\n", stringToOutput);
        fflush(stdout);
        char* fgetsRet = fgets(input, INPUT_SIZE, stdin);
        if (fgetsRet == NULL)
        {
            printf("Wrong format!\n");
            continue;
        }
        int inputLength = strlen(input) - 1;
        input[inputLength] = '\0';
        int sscanfRet = sscanf(input, "%d%n", &number, &position);
        if (position != inputLength)
        {
            printf("Wrong format!\n");
            continue;
        }
        if (pChecker && !pChecker(number))
        {
```

Продолжение листинга 1

```
            printf("Wrong format!\n");
            continue;
        }
        if (sscanfRet == 1) break;
        printf("Wrong format!\n");
    }
    return number;
}

bool ArraySizeInputChecker(int arraySize)
{
    return arraySize > 0;
}

bool AnyIntInputChecker(int _)
{
    return true;
}

typedef struct
{
    int* content;
    int size;
} DynArray;

void arrayCopy(DynArray* origin, DynArray* object)
{
    object->size = origin->size;
    object->content = (int*) malloc(object->size * sizeof(int));
    for (int i = 0; i < object->size; i++)
    {
        object->content[i] = origin->content[i];
    }
}

double arrayAverage(DynArray* object)
{
    double result = 0;
    for (int i = 0; i < object->size; i++)
    {
        result += object->content[i];
```

Окончание листинга 1

```
    }
    result /= object->size;
    return result;
}

double arrayMidpoint(DynArray* object)
{
    double result;
    result = 0;
    DynArray copy;
    arrayCopy(object, &copy);
    qsort(copy.content, copy.size, sizeof(int),
          (int (*)(const void*, const void*)) DefaultComp);
    if (object->size % 2 == 0)
    {
        result += (copy.content[(object->size / 2) - 1] +
                   copy.content[object->size / 2]);
        result/=2;
    }
    else
    {
        result = copy.content[object->size / 2];
}
    free(copy.content);
    return result;
}

int main()
{
    DynArray object;
    object.size = CycleInputInt("Enter size of array", ArraySizeInputChecker);
    object.content = (int*) malloc(object.size * sizeof(int));
    printf("Enter elements, one by one\n");
    for (int i = 0; i < object.size; i++)
    {
        object.content[i] = CycleInputInt("Enter next element",
                                          AnyIntInputChecker);
    }
    printf("Average: %0.2f\n", arrayAverage(&object));
    printf("Midpoint: %0.2f\n", arrayMidpoint(&object));
    free(object.content); }
```

5 Результаты работы программы

На следующем рисунке приведен скриншот с результатами работы программы.

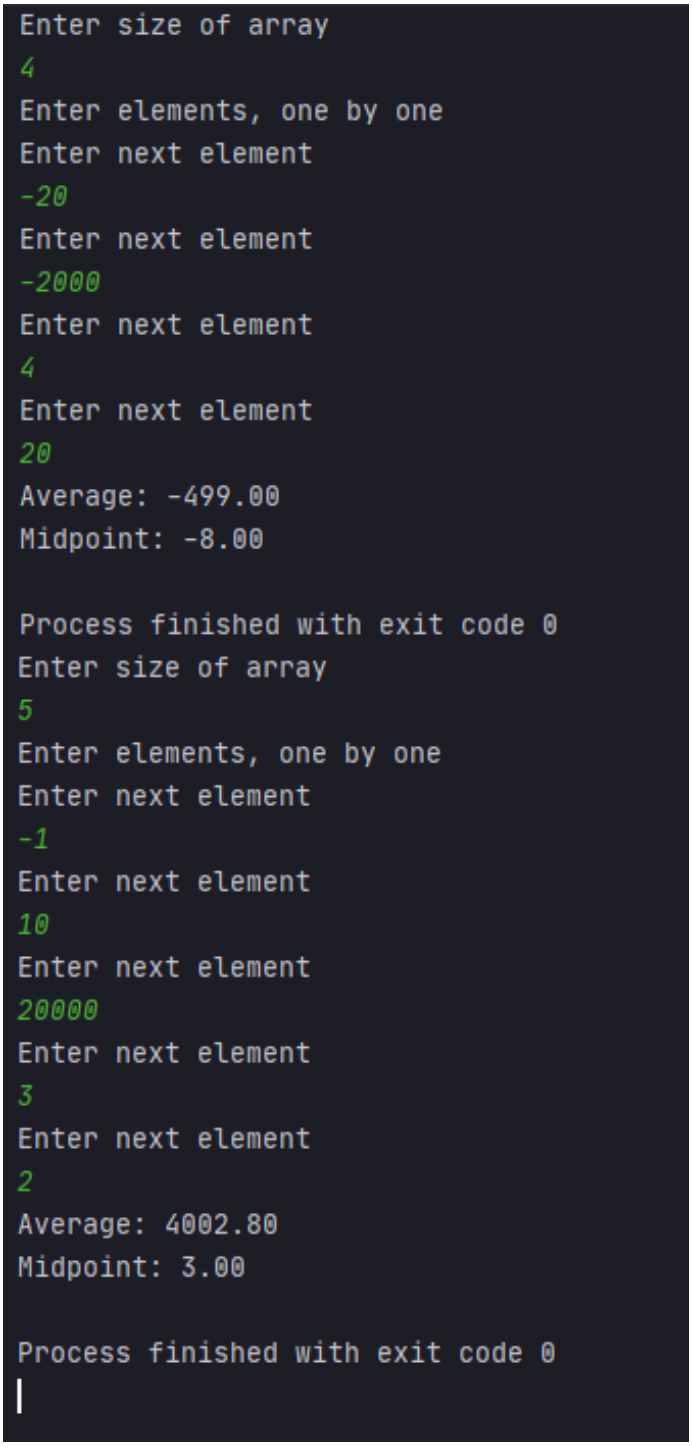

Рисунок 1 – Результаты работы программы