

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное образовательное учреждение высшего
образования**
**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ**
**УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №2

«Разветвляющиеся процессы обработки данных»

Студента 2 курса, ИСП-208

Каршибаева Жахонгира

Направление 09.02.07 – «Информационные системы и
программирование»

Руководитель:
Преподаватель
М.В. Валеев

Работа защищена

« _____
_____ »
“ ”
_____ 2022г.

Москва 2022

Оглавление

1.Номер первый	Ошибка! Закладка не определена.
1.1. Постановка задачи	Ошибка! Закладка не определена.
1.2. Структурные требование	Ошибка! Закладка не определена.
1.3. Описание работы	Ошибка! Закладка не определена.
1.4.1. Использование переменных	Ошибка! Закладка не определена.
1.4.2. Псевдокод программы	Ошибка! Закладка не определена.
1.4.3. Блок-схема алгоритма	Ошибка! Закладка не определена.
1.4.4. Листинг программной реализации	Ошибка! Закладка не определена.
1.4.5. Результаты	Ошибка! Закладка не определена.
1.4.6. Тесты	Ошибка! Закладка не определена.
2.Номер второй	Ошибка! Закладка не определена.
Постановка задачи	Ошибка! Закладка не определена.
2.2. Структурные требование	Ошибка! Закладка не определена.
2.3. Описание работы	Ошибка! Закладка не определена.
2.4.1. Использование переменных	Ошибка! Закладка не определена.
2.4.2. Псевдокод программы	Ошибка! Закладка не определена.
2.4.3. Блок-схема алгоритма	Ошибка! Закладка не определена.
2.4.4. Листинг программной реализации	Ошибка! Закладка не определена.
2.4.5. Результаты	Ошибка! Закладка не определена.
2.4.6. Тесты	Ошибка! Закладка не определена.
3.Номер третий	Ошибка! Закладка не определена.
3.1. Постановка задачи	Ошибка! Закладка не определена.
3.2. Структурные требование	Ошибка! Закладка не определена.
3.3. Описание работы	Ошибка! Закладка не определена.
3.4.1. Использование переменных	Ошибка! Закладка не определена.
3.4.2. Псевдокод программы	Ошибка! Закладка не определена.
3.4.3. Блок-схема алгоритма	Ошибка! Закладка не определена.
3.4.4. Листинг программной реализации	Ошибка! Закладка не определена.
3.4.5. Результаты	Ошибка! Закладка не определена.
3.4.6. Тесты	Ошибка! Закладка не определена.
4.Номер четвертый	Ошибка! Закладка не определена.
4.1. Постановка задачи	Ошибка! Закладка не определена.
4.2. Структурные требование	Ошибка! Закладка не определена.
3.3. Описание работы	Ошибка! Закладка не определена.
4.4.1. Использование переменных	Ошибка! Закладка не определена.
4.4.2. Псевдокод программы	Ошибка! Закладка не определена.
4.4.3. Блок-схема алгоритма	Ошибка! Закладка не определена.

4.4.4. Листинг программной реализации	Ошибка! Закладка не определена.
4.4.5. Результаты	Ошибка! Закладка не определена.
4.4.6. Тест	Ошибка! Закладка не определена.
5.Номер пяти	Ошибка! Закладка не определена.
5.1. Постановка задачи	Ошибка! Закладка не определена.
5.2. Структурные требование	Ошибка! Закладка не определена.
5.3. Описание работы	Ошибка! Закладка не определена.
5.4.1. Использование переменных	Ошибка! Закладка не определена.
5.4.2. Псевдокод программы	Ошибка! Закладка не определена.
5.4.3. Блок-схема алгоритма	Ошибка! Закладка не определена.
5.4.4. Листинг программной реализации	Ошибка! Закладка не определена.
5.4.5. Результаты	Ошибка! Закладка не определена.
5.4.6. Тесты	Ошибка! Закладка не определена.
Список литературы	Ошибка! Закладка не определена.

1. Постановка задачи №1

Найти максимальный и минимальный элементы массива

2. Структурные требования

Max_n>0. Задается как константа

3. Описание работы

3.1 Используемые переменные

i - переменная для цикла

min - минимальное значение массива

max - максимальное значение

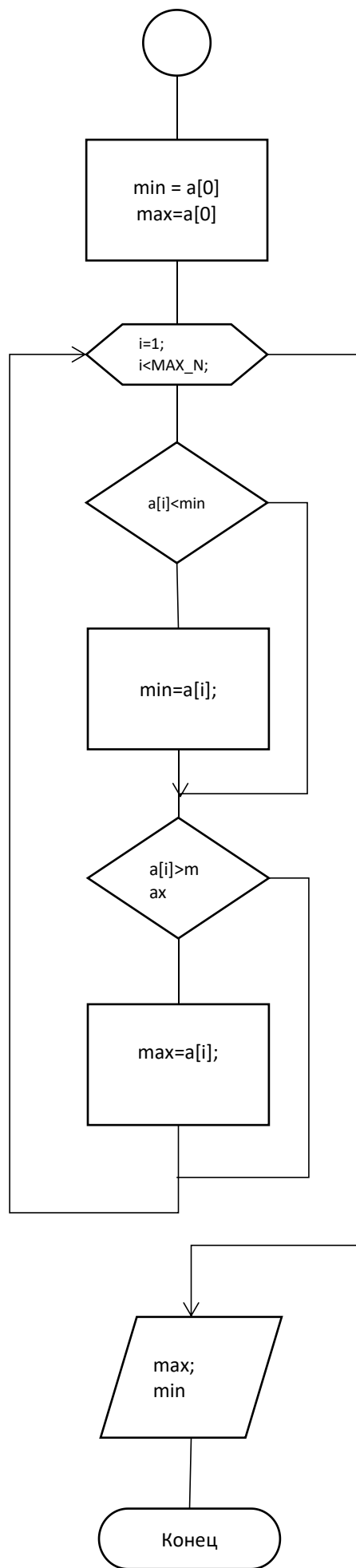
3.2 Псевдокод программы

int a[MAX_N] константа

for (i=0; i<10; i++) заполняем массив через цикл for до MAX_N, а затем присваиваем первой переменной значение мин и макс min=a[0];max=a[0]. Затем пробегаем по массиву и ищем минимальное значение, если находим то присваиваем новое значение мин if (a[i]<min){min=a[i];}; точно так же и с макс. И выводим значения.

3.3 Блок - схема алгоритма





3.4 Листинг программной реализации

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int MAX_N;
    scanf("%d", &MAX_N);
    int a[MAX_N];// объявление целочисленного массива а из 10
элементов
    int i, min, max;
    // ввод элементов массива с клавиатуры
    for (i=0; i<MAX_N; i++)
    {
        printf("Введите элемент массива %d\n", i);
        scanf("%d", &a[i]);
    }
    min =a[0];// присваиваем минимальному значению значение
первого элемента
    max=a[0];// присваиваем максимальному значению значение
первого элемента
    // пробегаем по всем элементам массива
    for (i=1; i<MAX_N; i++)
    {
        // если текущий элемент массива меньше текущего минимума то
        присваиваем его значение минимуму
        if (a[i]<min)
        {
            min=a[i];
        }
        // если текущий элемент массива больше текущего максимума то
        присваиваем его значение максимуму
        if (a[i]>max)
        {
            max=a[i];
        }
    }
    // выводим значения на экран
    printf("Максимальное значение %d\n",max);
    printf("Минимальное значение %d\n",min);
    return 0; }
```

3.5 Результат

```
Введите элемент массива 1
4
Введите элемент массива 2
5
Введите элемент массива 3
6
Введите элемент массива 4
3
Максимальное значение 6
Минимальное значение 1
```

3.6 Тесты

MAX N = 5

1 = 5	max = 78
2 = 4	
3 = 78	
4 = 9	min = 4
5 = 64	

1 = 1	max = 79
2 = 4	
3 = 8	
4 = 79	min = 1
5 = 64	

1. Постановка задачи №2

Переставить местами максимальный и минимальный элементы

2. Структурные требования

Максимальная размерность массива MAX_N задается глобальной константой

Текущая размерность массива n вводится в программе через оператор

Ввода

MAX_n > 0

3. Описание работы

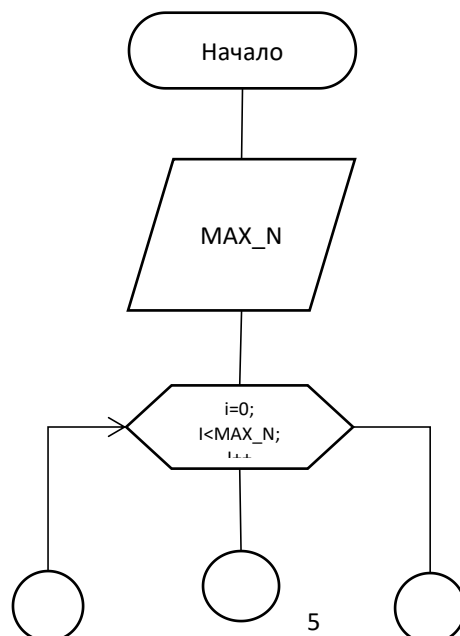
3.1 Используемые переменные

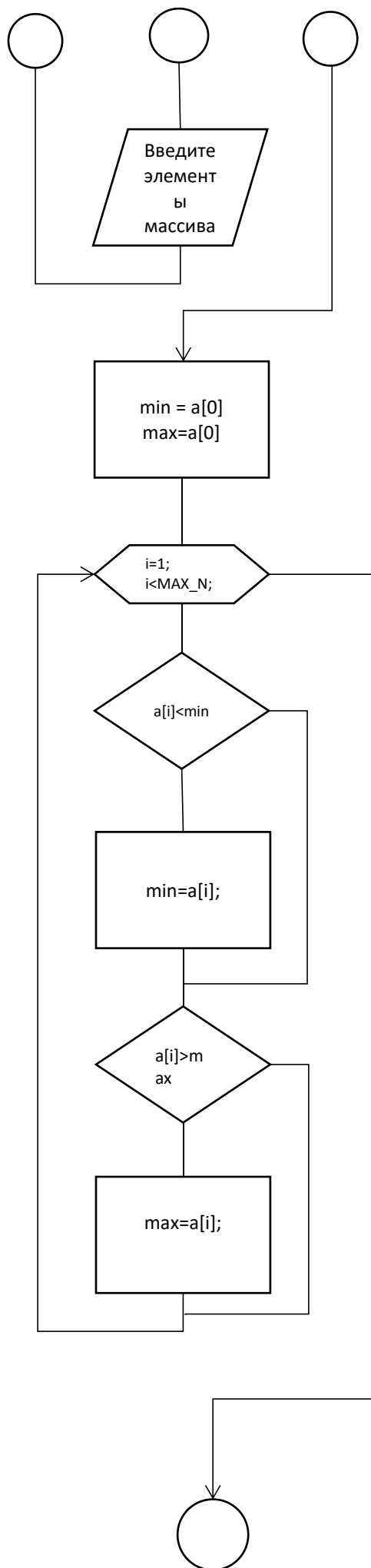
i, max, min, P

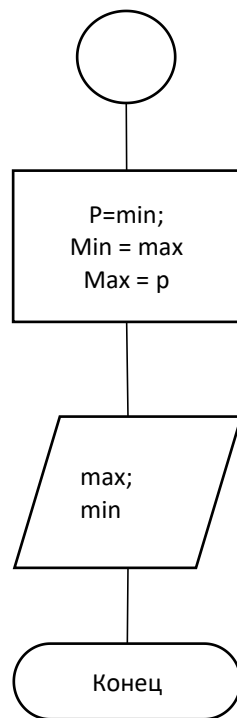
3.2 Псевдокод программы

Начало. Вводим MAX_N как константу и вводим его, будет количеством. После чего мы заполняем массив через клавиатуру и начинаем проверку массива. присваиваем первой переменной значение мин и макс $\min = a[0]; \max = a[0]$. Затем пробегаем по массиву и ищем минимальное значение, если находим то присваиваем новое значение мин $\text{if } (a[i] < \min) \{ \min = a[i]; \}$; точно так же и с макс. И выводим значения. После чего присваиваем мин значени P, а макс значение мин. Выводим значени макс и мин. Конец

3.3 Блок схема алгоритма







3.4 Листинг программной реализации

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int MAX_N;
    scanf("%d", &MAX_N);
    int a[MAX_N]; // объявление целочисленного массива a из 10
элементов
    int i, min, max, P;
    // ввод элементов массива с клавиатуры
    for (i=0; i<MAX_N; i++)
    {
        printf("Введите элемент массива %d\n", i);
        scanf("%d", &a[i]);
    }
    min =a[0]; // присваиваем минимальному значению значение
первого элемента
    max=a[0]; // присваиваем максимальному значению значение
первого элемента
    // пробегаем по всем элементам массива
    for (i=1; i<MAX_N; i++)
    {
        // если текущий элемент массива меньше текущего минимума то
        присваиваем его значение минимуму
        if (a[i]<min){min=a[i];};
    }
}
  
```

```
// если текущий элемент массива больше текущего максимума то
присваиваем его значение максимуму
if (a[i]>max){max=a[i];}
}
P=min; //значение мин сохраняем в P потом нам это понадобится
min = max; //значению мин присваиваем значение макс
max = P; //значение макс присваиваем P, точнее мин
// выводим значения на экран
printf("Максимальное значение %d\n",max);
printf("Минимальное значение %d\n",min);
return 0;
}
```

3.5 Результаты

```
5
Введите элемент массива 0
1
Введите элемент массива 1
4
Введите элемент массива 2
7
Введите элемент массива 3
8
Введите элемент массива 4
9
Максимальное значение 1
Минимальное значение 9
```

3.6 Тесты

```
4
Введите элемент массива 3
7
Введите элемент массива 4
87
Введите элемент массива 5
98
Введите элемент массива 6
65
Введите элемент массива 7
54
Максимальное значение 3
Минимальное значение 98
```

1. Постановка задачи №3

Найти количество различных элементов в массиве. Вывести значения неповторяющихся элементов.

2. Структурные требования

Максимальная размерность массива MAX_N задается глобальной константой

Текущая размерность массива n вводится в программе через оператор Ввода $n > 0$

3. Описание работы

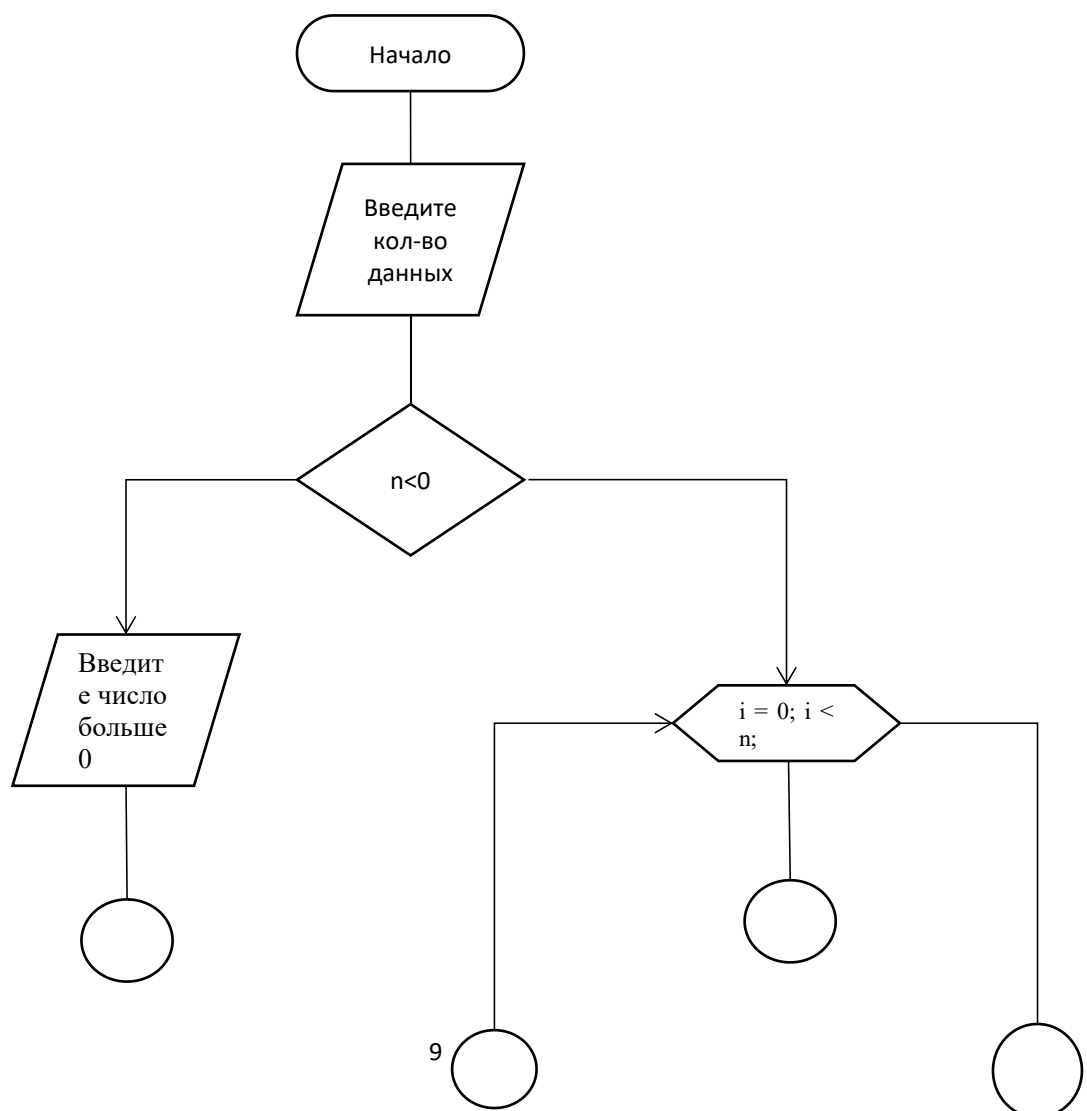
3.1 Используемые переменные

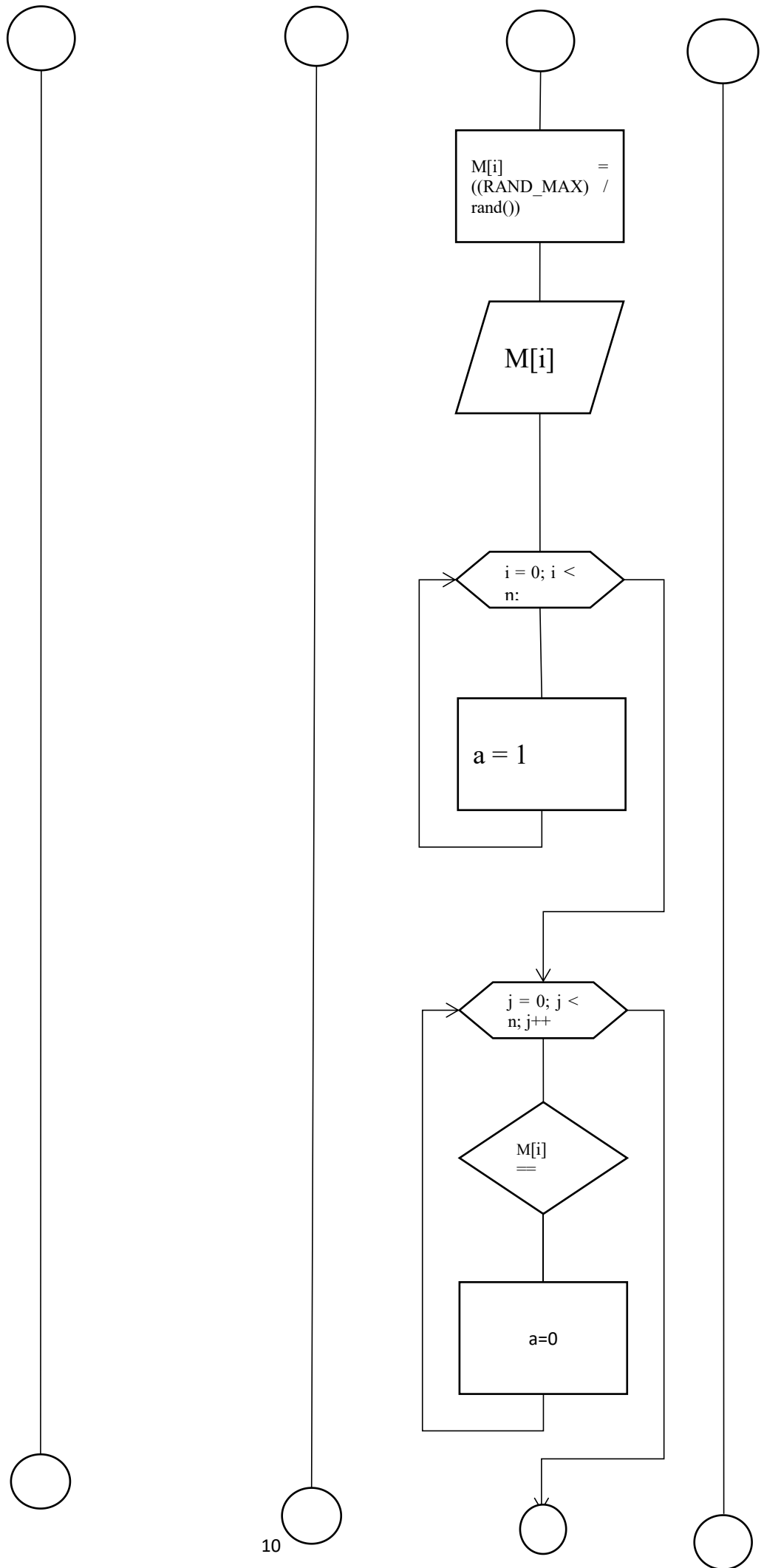
M, a, c, n, I, j

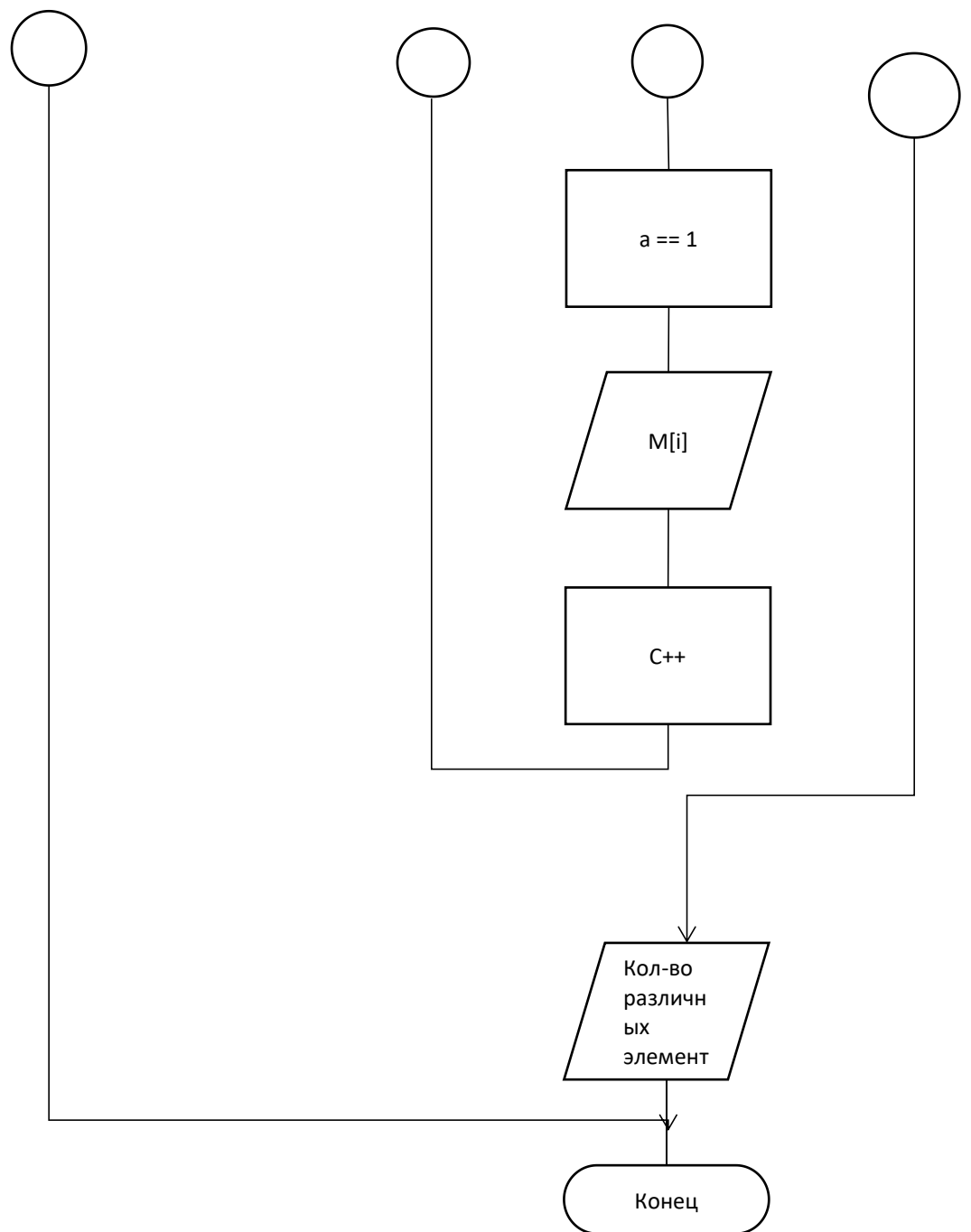
3.2 Псевдокод программы

Начало. Объявляем переменные и заполняем массив рандом $M[i] = ((\text{RAND_MAX}) / \text{rand}())$. Затем пробегаем по массиву в поисках различных. Проверяем массив I на наличие различных элементов. Затем выводим сам массив и количество различных.

3.3 Блок схема







3.4 Листинг программной реализации

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

```

```

int main()
{
    srand(time(NULL));
    int* M, a, c = 0, n;
    printf(" Введите кол-во элементов: \n");
    scanf("%i", &n);
    if (n < 1) printf("Введите число больше 0\n");
    else
    {

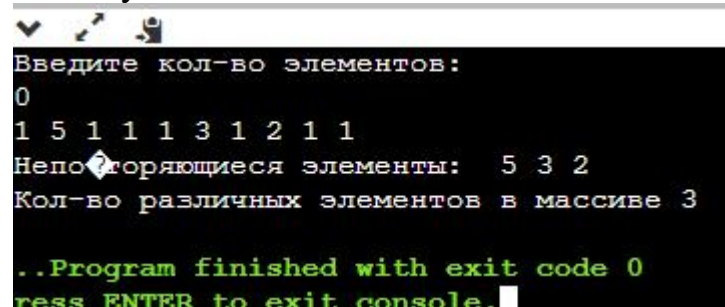
```

```

for (int i = 0; i < n; i++)
{
M[i] = ((RAND_MAX) / rand());
printf(" %i", M[i]);
}
printf("\n Неповторяющиеся элементы: ");
for (int i = 0; i < n; i++)
{
a = 1;
for (int j = 0; j < n; j++)
{
if ((M[i] == M[j]) && (i != j))
{
a = 0;
}
}
if (a == 1)
{
printf(" %i", M[i]);
c++;
}
}
printf("\n Кол-во различных элементов в массиве %i", c);
}
return 0;
}

```

3.5 Результат

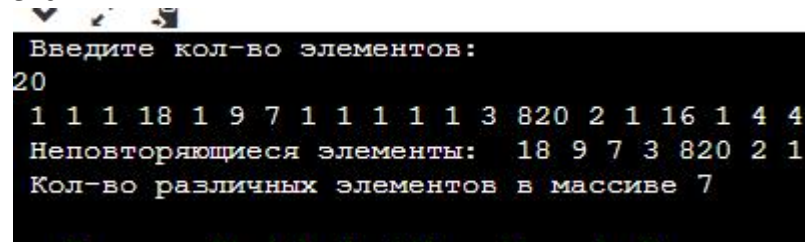


```

Введите кол-во элементов:
0
1 5 1 1 1 3 1 2 1 1
Неповторяющиеся элементы: 5 3 2
Кол-во различных элементов в массиве 3
..Program finished with exit code 0
press ENTER to exit console.

```

3.6 Тесты



```

Введите кол-во элементов:
20
1 1 1 18 1 9 7 1 1 1 1 3 820 2 1 16 1 4 4
Неповторяющиеся элементы: 18 9 7 3 820 2 1
Кол-во различных элементов в массиве 7
..Program finished with exit code 0
press ENTER to exit console.

```

1. Постановка задачи №4

В одномерном массиве, состоящем из n целых элементов, вычислить: минимальный по модулю элемент массива;

2. Структурные требования

Максимальная размерность массива MAX_N задается глобальной константой

Текущая размерность массива n вводится в программе через оператор Ввода $n > 0$

3. Описание работы

3.1 Используемые переменные

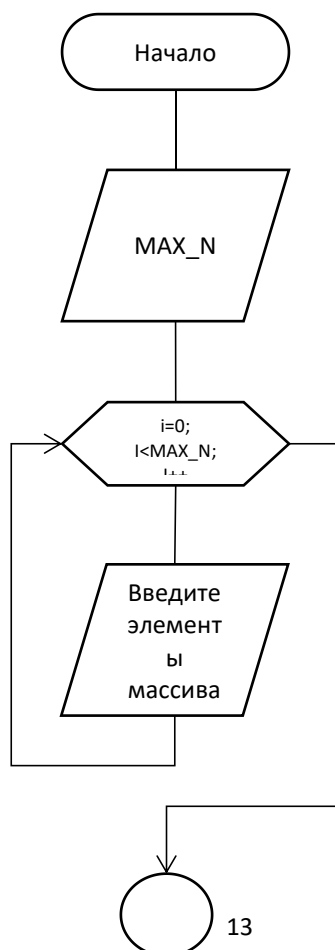
i , min , max

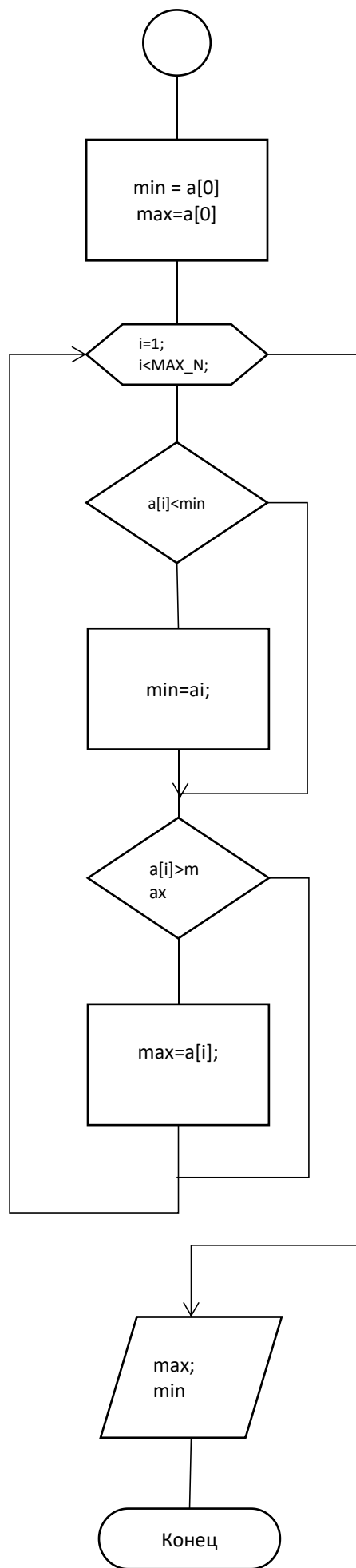
3.2 Псевдокод программы

`int a[MAX_N]` константа

`for (i=0; i<10; i++)` заполняем массив через цикл `for` до MAX_N , а затем присваиваем первой переменной значение мин и макс `min=a[0];max=a[0]`. Затем пробегаем по массиву и ищем минимальное значение, если находим то присваиваем новое значение мин `if (a[i]<min){min=a[i];}`; точно так же и с макс. И выводим значения.

3.3 Блок схема алгоритма





3.4 Листинг программной реализации

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int MAX_N;
    scanf("%d", &MAX_N);
    int a[MAX_N]; // объявление целочисленного массива a из 10
элементов
    int i, min, max;
    // ввод элементов массива с клавиатуры
    for (i=0; i<MAX_N; i++)
    {
        printf("Введите элемент массива %d\n", i);
        scanf("%d", &a[i]);
    }
    min =a[0]; // присваиваем минимальному значению значение первого
элемента
    max=a[0]; // присваиваем максимальному значению значение
первого элемента
    // пробегаем по всем элементам массива
    for (i=1; i<MAX_N; i++)
    {
        // если текущий элемент массива меньше текущего минимума то
        присваиваем его значение минимуму
        if (a[i]<min)
        {
            min=i;
        }
        // если текущий элемент массива больше текущего максимума то
        присваиваем его значение максимуму
        if (a[i]>max)
        {
            max=a[i];
        }
    }
    // выводим значения на экран
    printf("Максимальное значение %d\n",max);
    printf("Минимальное значение %d\n",min);
    return 0;
}
```

3.5 Результат

```
5
Введите элемент массива 3
7
Введите элемент массива 4
8
Максимальное значение 8
Минимальное значение 1
```

3.6 Тесты

```
4
Введите элемент массива 3
1
Введите элемент массива 4
2
Максимальное значение 8
Минимальное значение 4
```

```
2
Введите элемент массива 8
1
Введите элемент массива 9
5
Максимальное значение 87
Минимальное значение 9
```

1. Постановка задачи №5

В одномерном массиве, состоящем из n целых элементов, вычислить: сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

2. Структурные требования

Максимальная размерность массива MAX_N задается глобальной константой

Текущая размерность массива n вводится в программе через оператор Ввода $MAX_N > 0$

3. Описание работы

3.1 Используемые переменные

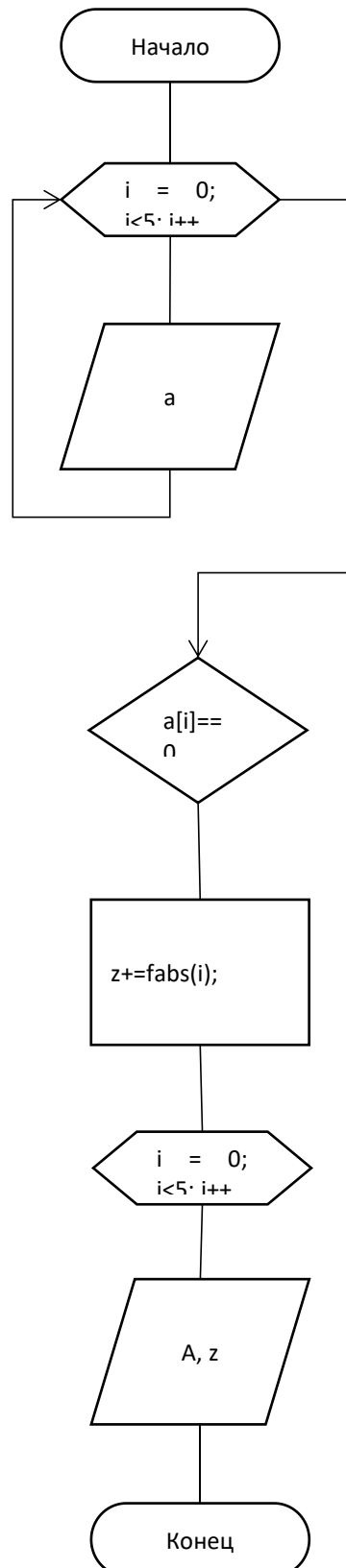
a, z, i

3.2 Псевдокод крограммы

Начало

Объявляем массив из 5 `int a[5]`; элементов. Затем заполняем массив (`i = 0; i < 5; i++`) через цикл после чего при помощи `if` ищем ноль и плюсуем следующие

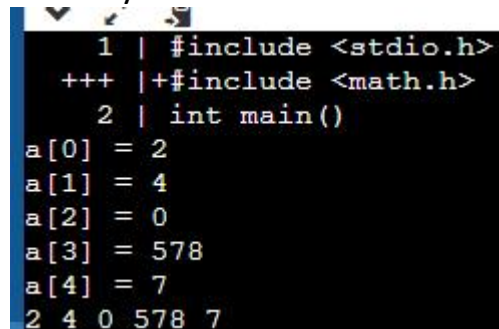
3.3 Блок схема алгоритма



3.4 Листинг программной реализации

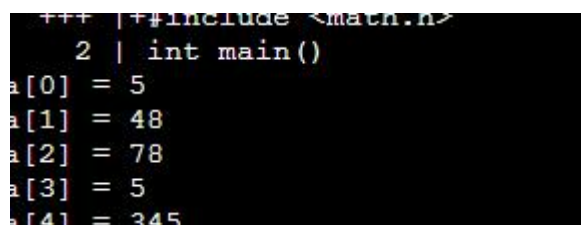
```
#include <stdio.h>
int main()
{
    int a[5]; // объявлен массив a из 5 элементов
    int i, z;
    // Ввод элементов массива
    for (i = 0; i < 5; i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]); // &a[i] - адрес i-го элемента массива
    }
    if (a[i] == 0)
    {
        z += fabs(i);
    }
    // Вывод элементов массива
    for (i = 0; i < 5; i++)
        printf("%d ", a[i]); // пробел в формате печати обязателен
    //getchar(); getchar();
    return 0;
}
```

3.5 Результат



```
1 | #include <stdio.h>
+++ |+#include <math.h>
2 | int main()
a[0] = 2
a[1] = 4
a[2] = 0
a[3] = 578
a[4] = 7
2 4 0 578 7
```

3.6 Тесты



```
+++ |+#include <math.h>
2 | int main()
a[0] = 5
a[1] = 48
a[2] = 78
a[3] = 5
a[4] = 345
```

Список литературы

1. Конова Е.А., Поллак Г.А. Алгоритмы и программы. Язык С++: Учебное пособие. - 2-е изд., стер. - СПб.: Издательство "Лань", 2017. - 384 с.
2. Седжвик Роберт. Алгоритмы на С++.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2011. - 1056 с.
3. Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. - СПб.: Питер, 2015. - 928 с.
4. Орлов С.А. Теория и практика языков программирования: Учебник для вузов. Стандарт 3-го поколения. - СПб.: Питер, 2014. - 688 с.
5. Павловская Т.А. С/С++. Процедурное и объектно-ориентированное программирование: Учебник для вузов. Стандарт 3-го поколения. - СПб.: Питер, 2015. - 496 с.
6. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник для вузов. - СПб.: Питер, 2014. - 432 с.
7. Плаксин М.А. Тестирование и отладка программ - для профессионалов будущих и настоящих. - М.: БИНОМ. Лаборатория знаний, 2013. - 167 с.