



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)


ФАКУЛЬТЕТ ИУК «Информатика и управление»
КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №4


«Обработка одномерных массивов»

ДИСЦИПЛИНА: «Высокоуровневое программирование»

Выполнил: студент гр. ИУК4-11Б


(подпись) (Суриков Н.С)
(Ф.И.О.)

Проверил:


(подпись) (Пчелинцева Н.И.)
(Ф.И.О.)

Дата сдачи (защиты):

26 20.11.23

Результаты сдачи (защиты):

- Балльная оценка:

50

- Оценка:

хорошо

Цель работы: изучение приемов работы с одномерными массивами средствами языка C++.

Задачи:

1. Разработать программы с использованием массивов.
2. Познакомиться с операциями обработки элементов в массивах.
3. Изучить типовые алгоритмы решения задач с использованием массивов.

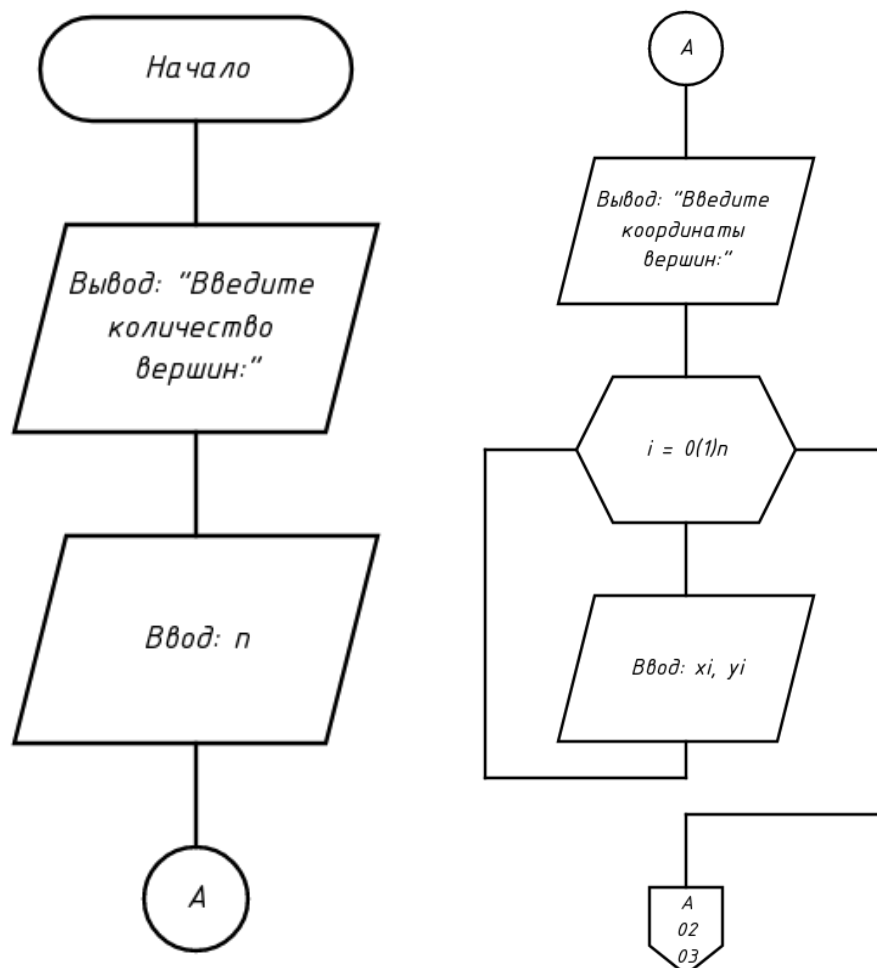
Вариант 11

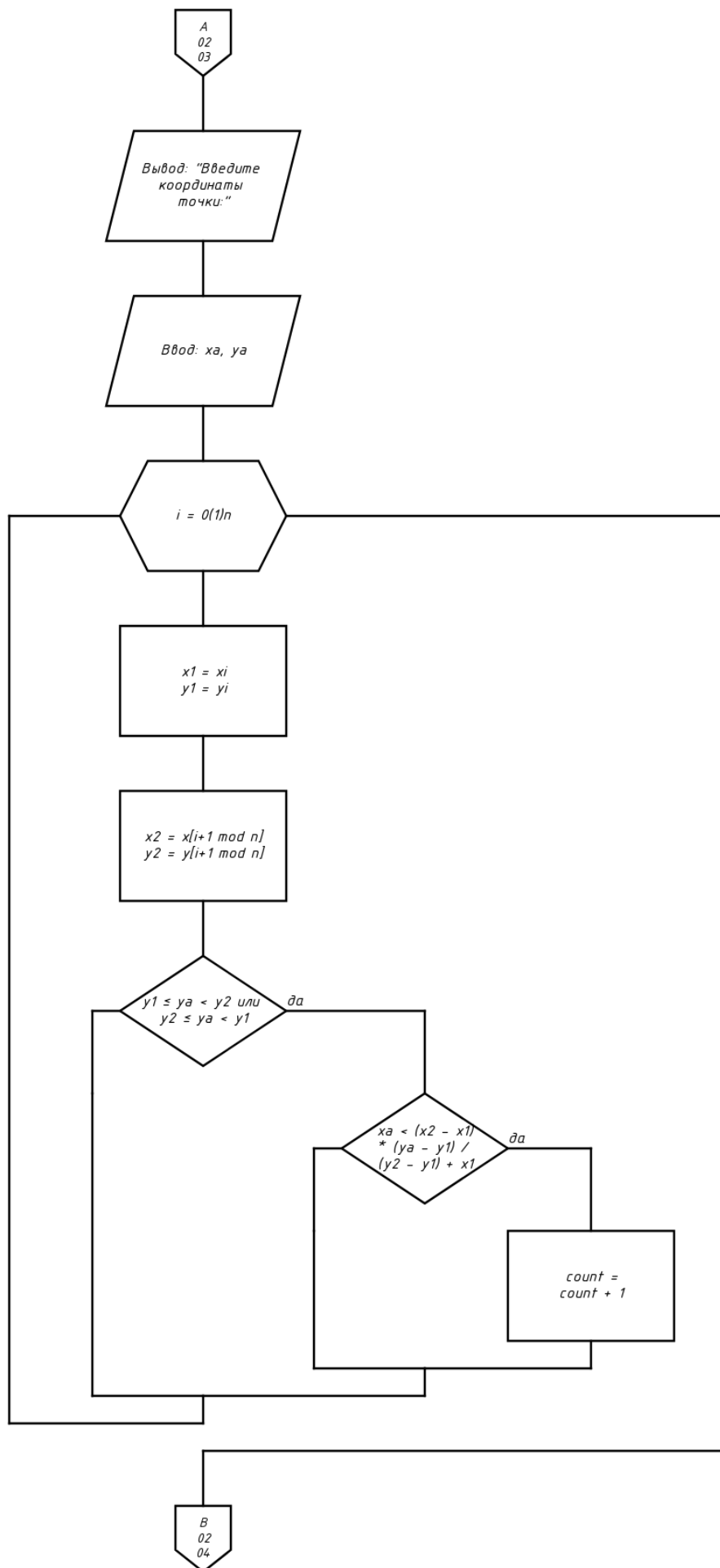
Задание 1:

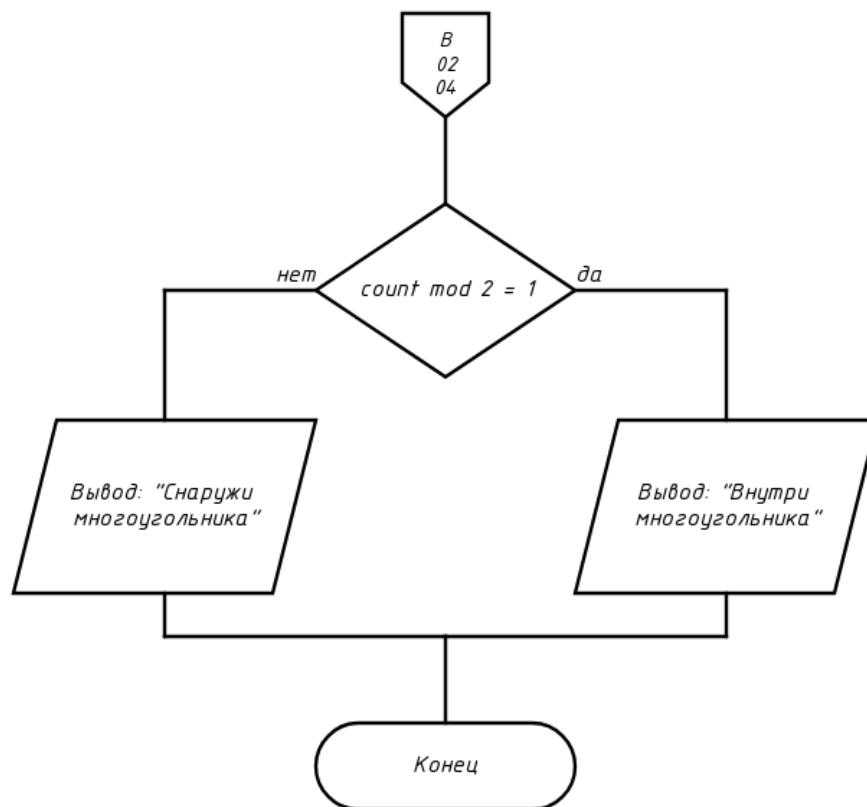
Формулировка:

Выпуклый многоугольник задан координатами своих вершин. Вершины упорядочены. Определить, находится ли точка a с координатами x_a , y_a внутри многоугольника.

Блок схема:







Листинг программы на C++:

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n{};
    cout << "Введите количество вершин: ";
    cin >> n;

    int x[n], y[n];
    cout << "Введите координаты вершин: " << endl;
    for (int i = 0; i < n; i++)
    {
        cin >> x[i] >> y[i];
    }

    int xa, ya;
    cout << "Введите координаты точки: ";
    cin >> xa >> ya;

    int count = 0;
    for (int i = 0; i < n; i++)
    {
        int x1 = x[i];
        int y1 = y[i];
```

```

int x2 = x[(i + 1) % n];
int y2 = y[(i + 1) % n];

if (((y1 <= ya && ya < y2) || (y2 <= ya && ya < y1)) &&
(xa < (x2 - x1) * (ya - y1) / (y2 - y1) + x1))
{
count++;
}
}

if (count % 2 == 1)
{
cout << "Внутри многоугольника" << endl;
}
else
{
cout << "Снаружи многоугольника" << endl;
}

return 0;
}

```

Результат выполнения:

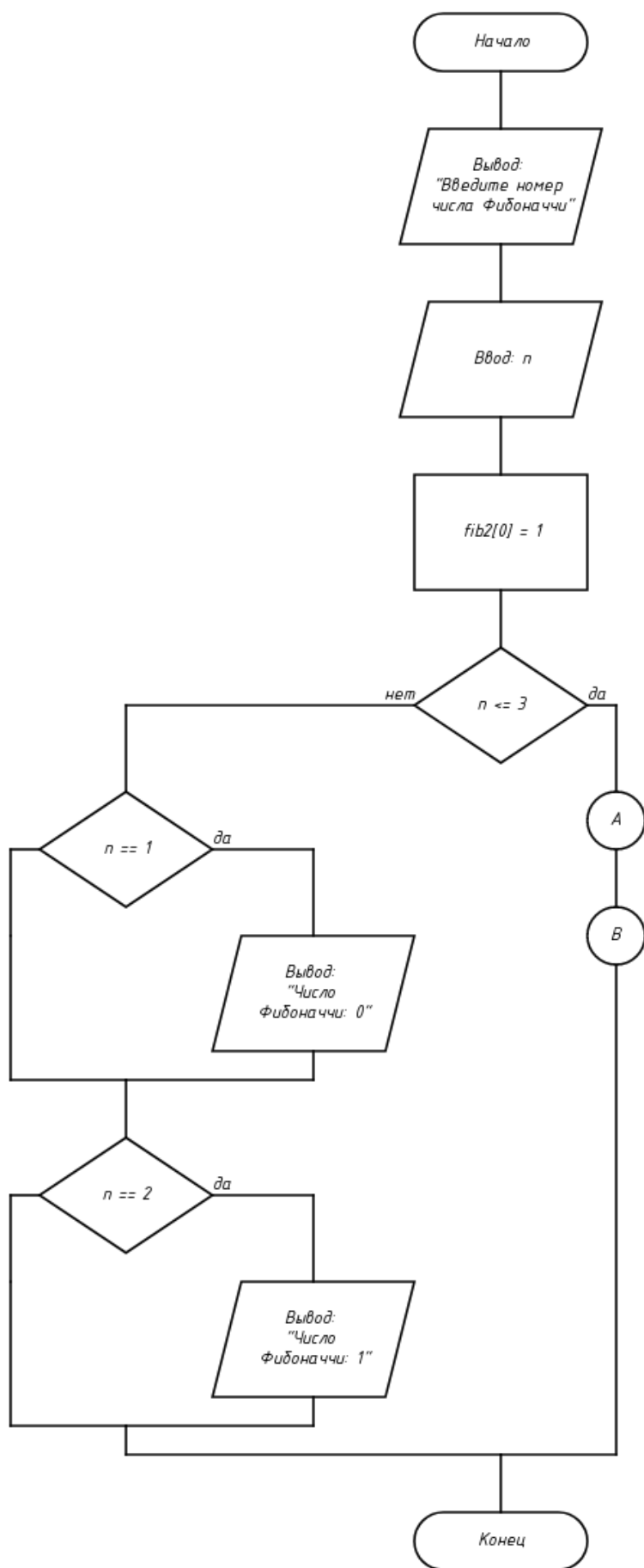
| | | | | |
|----------------------------|--------------------------------------|-----------------------------------|---|---|
| <i>Входные данные</i> | 4 0 0 0 4 4 4 4 0 2 2 | 3 0 0 0 4 4 0 2 5 | 5 0 0 0 4 4 4 4 2 2 0 1 1 | 6 0 0 0 4 2 5 4 4 4 0 2 -1 5 3 |
| <i>Вывод программы</i> | <i>Внутри многоугольника</i> | <i>Снаружи многоугольника</i> | <i>Внутри многоугольника</i> | <i>Снаружи многоугольника</i> |

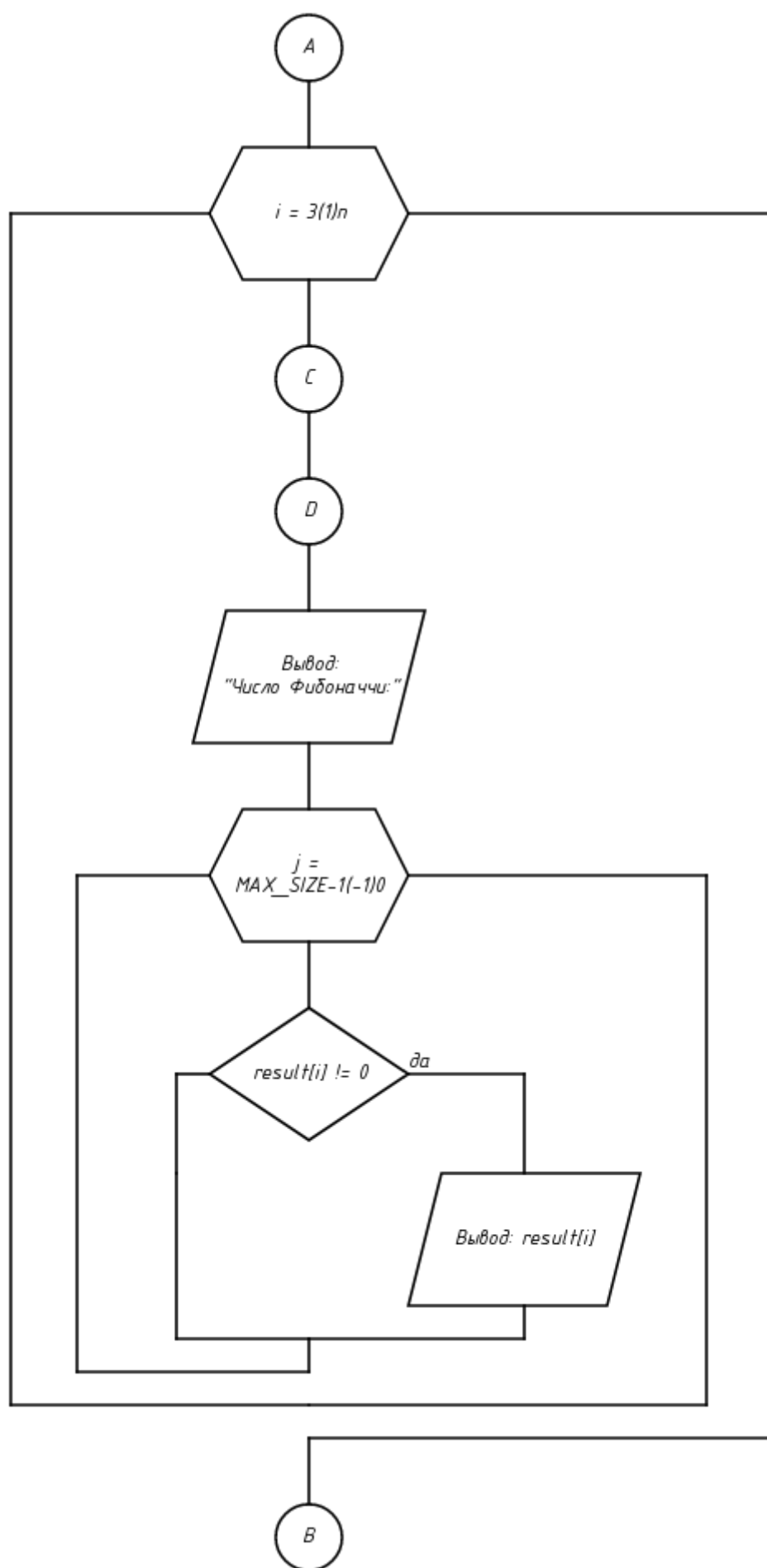
Задание 2:

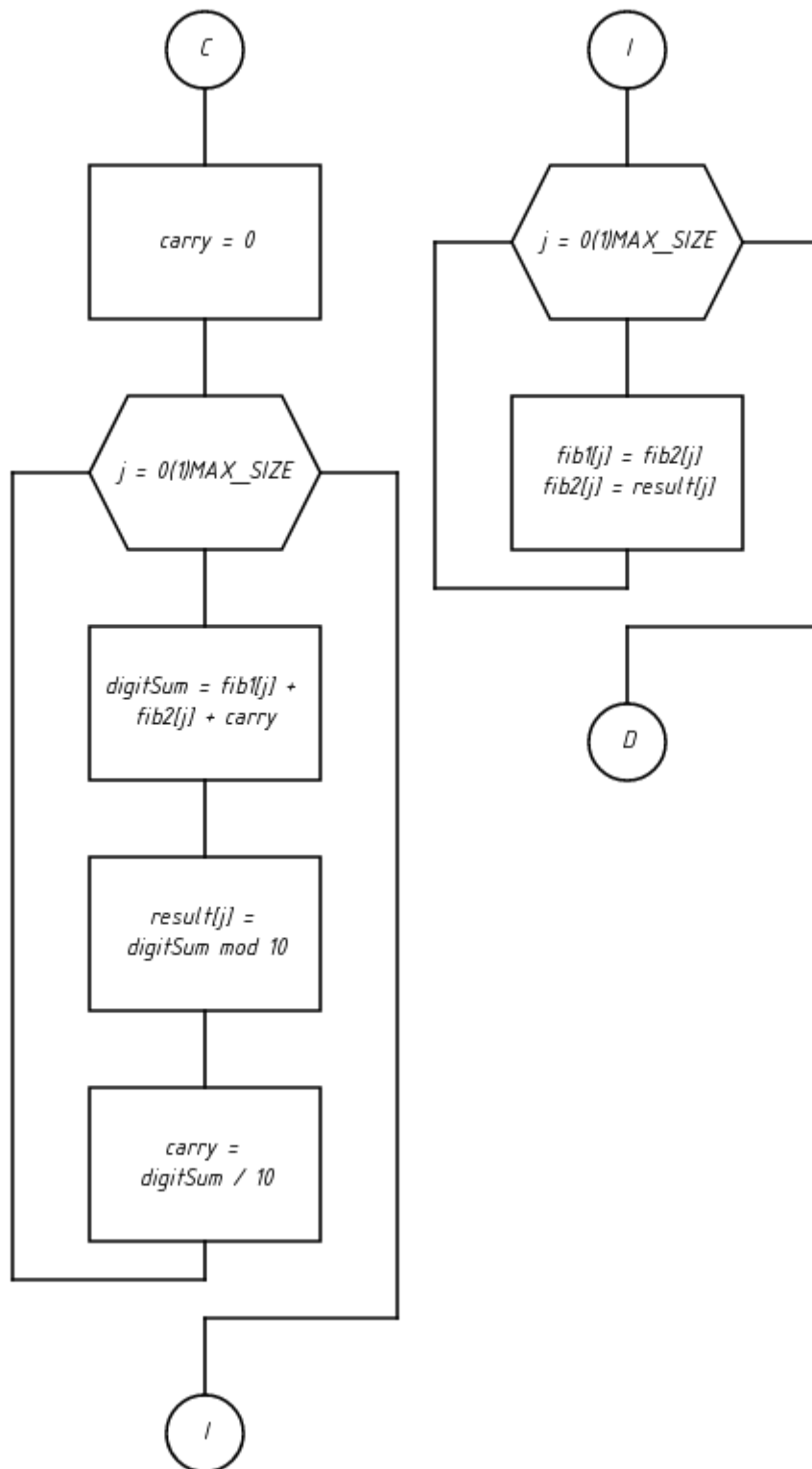
Формулировка:

Вычислить число Фибоначчи с номером n (каждое число Фибоначчи представить в виде целочисленного массива десятичных цифр).

Блок схема:







Листинг программы на C++:

```

#include <iostream>
using namespace std;

const int MAX_SIZE = 30; // Максимальное количество цифр в числе Фибоначчи
  
```



```

int main()
{
    int n;
    cout << "Введите номер числа Фибоначчи: ";
    cin >> n;

    int fib1[MAX_SIZE]{0}; // Первое число Фибоначчи
    int fib2[MAX_SIZE]{0}; // Второе число Фибоначчи
    int result[MAX_SIZE]{0}; // Результат

    fib2[0] = 1; // Инициализация второго числа Фибоначчи как 1
    if (n >= 3)
    {
        for (int i = 3; i <= n; i++)
        {
            // Сложение двух чисел Фибоначчи
            int carry = 0; // Перенос разряда

            for (int j = 0; j < MAX_SIZE; j++)
            {
                int digitSum = fib1[j] + fib2[j] + carry; // Сумма цифр на текущем разряде
                result[j] = digitSum % 10; // Записываем последнюю цифру суммы
                carry = digitSum / 10; // Вычисляем перенос разряда
            }
            // Обновляем значения чисел Фибоначчи для следующей итерации
            for (int j = 0; j < MAX_SIZE; j++)
            {
                fib1[j] = fib2[j];
                fib2[j] = result[j];
            }
        }

        cout << "Число Фибоначчи: ";
        for (int i = MAX_SIZE - 1; i >= 0; i--)
        {
            if (result[i] != 0)
            {
                cout << result[i];
            }
        }
        cout << endl;
    }
    else
    {
        if (n == 1)
            cout << "Число Фибоначчи: 0" << endl;
        if (n == 2)
            cout << "Число Фибоначчи: 1" << endl;
    }
}

```

Результат выполнения:

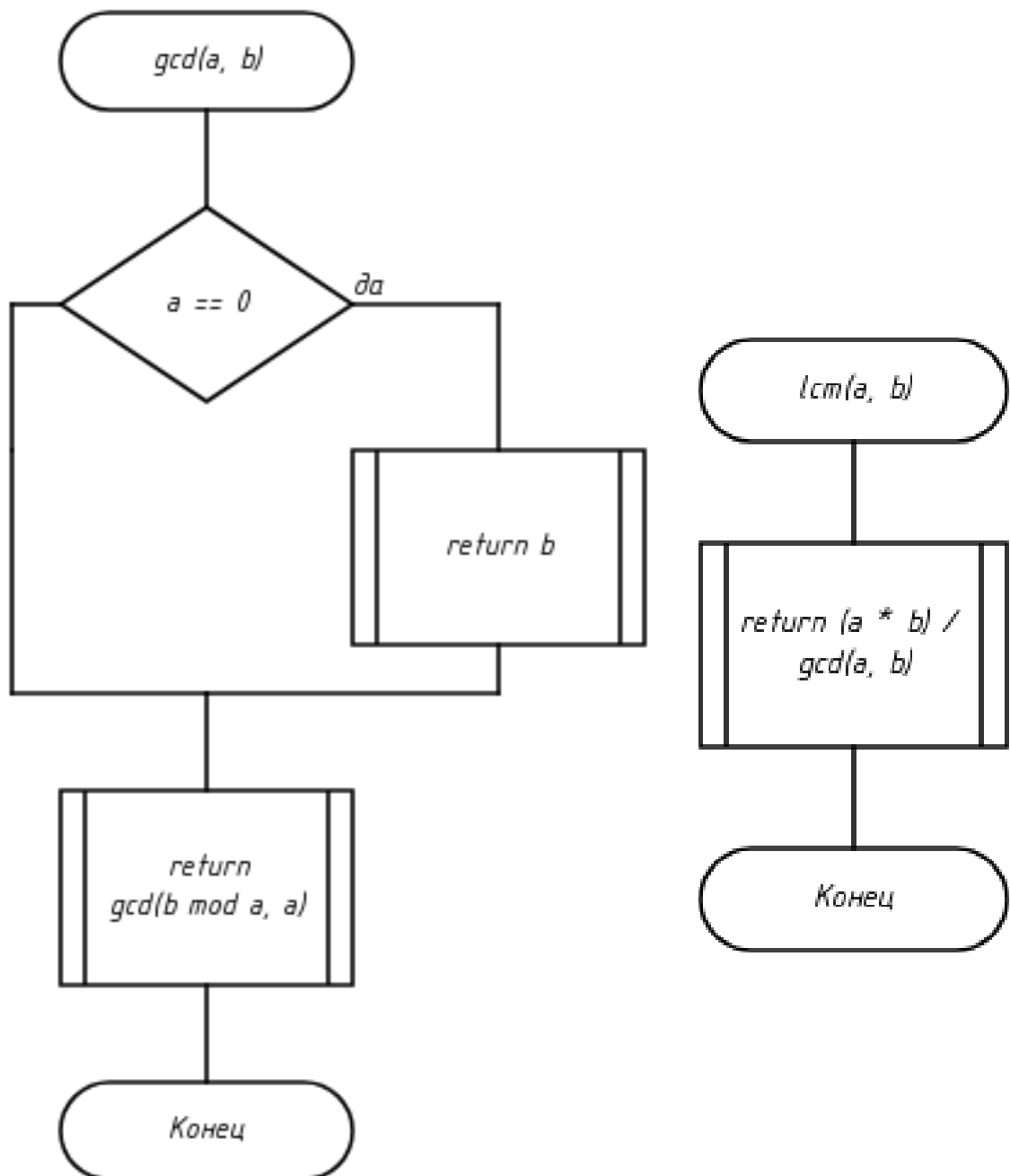
| | | | | | | | |
|------------------------|----------|----------|----------|----------|----------|----------|-----------|
| <i>Входные данные</i> | <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> | <i>5</i> | <i>6</i> | <i>10</i> |
| <i>Вывод программы</i> | <i>0</i> | <i>1</i> | <i>1</i> | <i>2</i> | <i>3</i> | <i>5</i> | <i>34</i> |

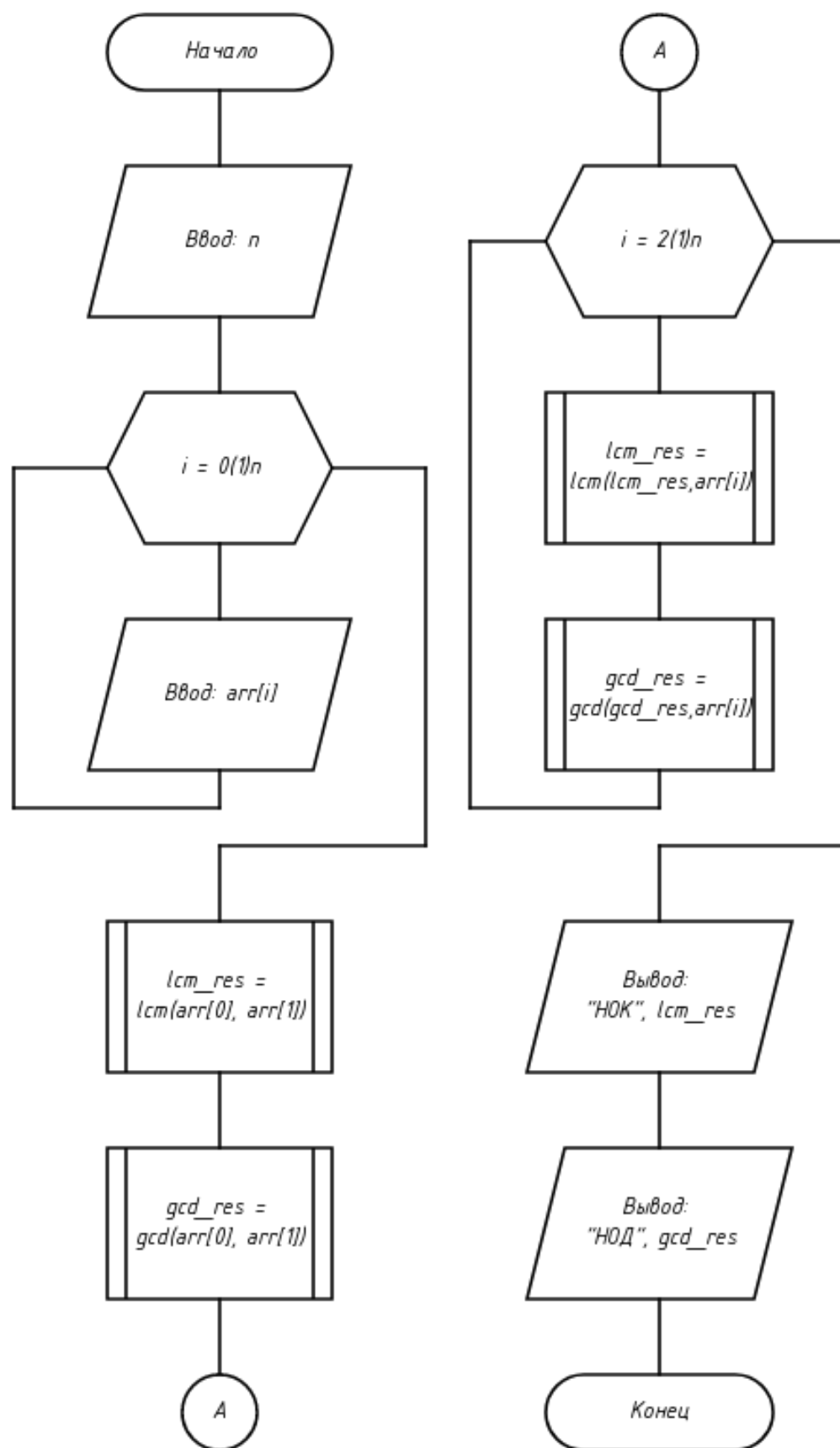
Задание 3:

Формулировка:

Найти наименьшее общее кратное (НОК) и наибольший общий делитель (НОД) для n введенных натуральных чисел.

Блок схема:





Листинг программы на C++:

```
#include <iostream>
#include <algorithm>
using namespace std;

// Функция для нахождения НОД двух чисел
int gcd(int a, int b) {
    if (a == 0) {
        return b;
    }
    return gcd(b % a, a);
}

// Функция для нахождения НОК двух чисел
int lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}

int main() {
    int n;
    cin >> n;

    int arr[n];
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int lcm_res = lcm(arr[0], arr[1]);
    int gcd_res = gcd(arr[0], arr[1]);

    for (int i = 2; i < n; i++) {
        lcm_res = lcm(lcm_res, arr[i]);
        gcd_res = gcd(gcd_res, arr[i]);
    }

    cout << "НОК: " << lcm_res << endl;
    cout << "НОД: " << gcd_res << endl;
}
```

Результат выполнения:

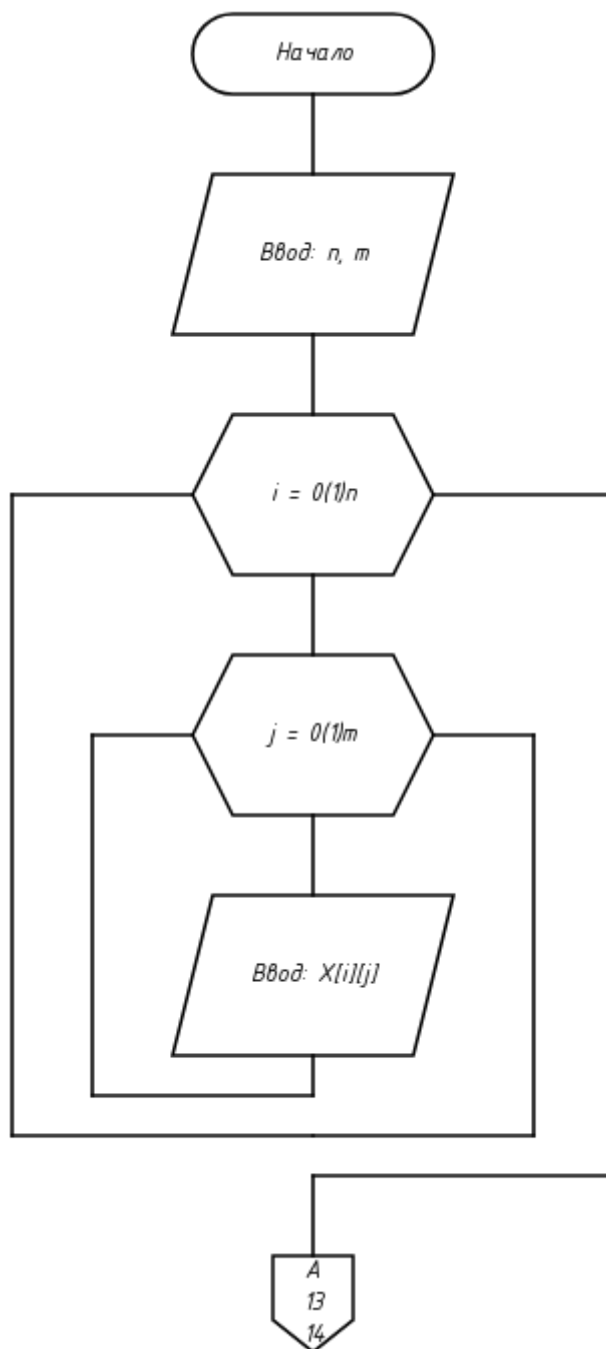
| | | | | | |
|-----------------|--------------------|-------------------|---------------------|-------------------|-------------------------|
| Входные данные | 5 2 4 6 8 10 | 4 3 6 9 12 | 3 7 11 13 | 2 5 7 | 6 4 8 12 16 20 24 |
| Вывод программы | НОК: 120 НОД: 2 | НОК: 36 НОД: 3 | НОК: 1001 НОД: 1 | НОК: 35 НОД: 1 | НОК: 240 НОД: 4 |

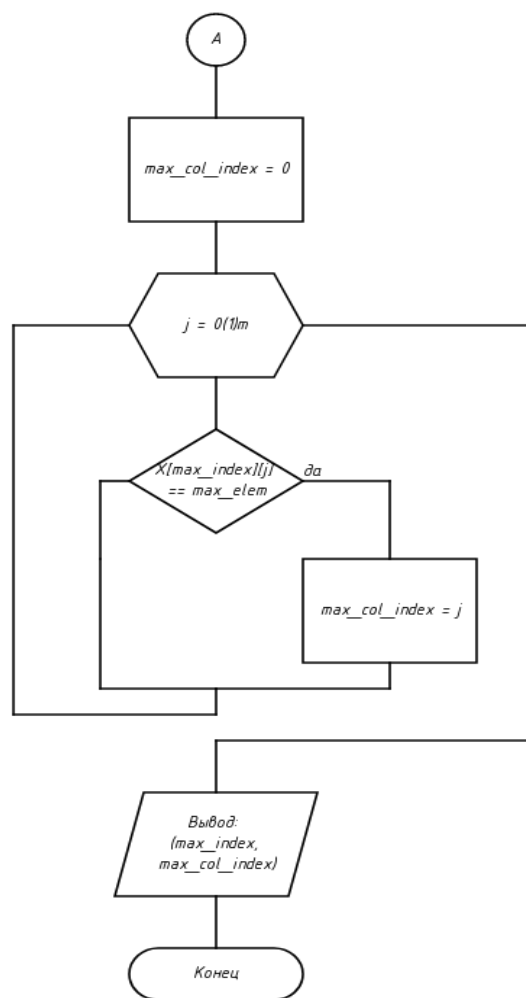
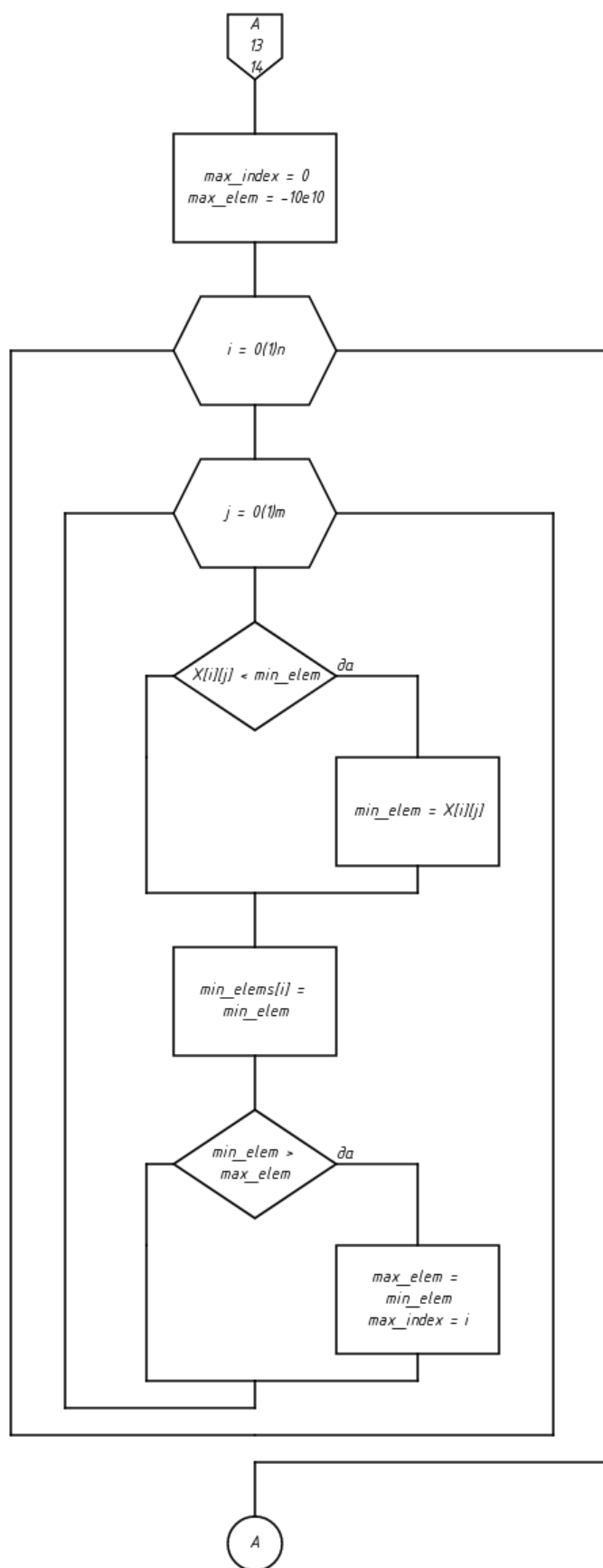
Задание 4:

Формулировка:

В двумерном массиве $X(n, m)$ все числа различны. В каждой строке находится минимальный элемент, затем среди этих чисел находится максимальное. Напечатать индексы (номер строки и номер столбца) этого элемента.

Блок схема:





Листинг программы на C++:

```
#include <iostream>
#include <climits>
using namespace std;

int main()
{
    int n, m;
    cin >> n >> m;

    int X[n][m];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> X[i][j];
        }
    }

    int min_elems[n]; // Массив для хранения минимальных элементов каждой строки
    int max_index = 0; // Индекс максимального элемента среди минимальных
    // элементов
    int max_elem = -10e10; // Максимальный элемент среди минимальных элементов

    // Находим минимальные элементы каждой строки и запоминаем индекс
    // максимального элемента
    for (int i = 0; i < n; i++)
    {
        int min_elem = 10e10;
        for (int j = 0; j < m; j++)
        {
            if (X[i][j] < min_elem)
            {
                min_elem = X[i][j];
            }
        }
        min_elems[i] = min_elem;
        if (min_elem > max_elem)
        {
            max_elem = min_elem;
            max_index = i;
        }
    }

    // Находим индекс столбца максимального элемента
    int max_col_index = 0;
    for (int j = 0; j < m; j++)
    {
        if (X[max_index][j] == max_elem)
        {
            max_col_index = j;
            break;
        }
    }

    cout << "(" << max_index << ", " << max_col_index << ")" << endl;

    return 0;
}
```

Результат выполнения:

| | | | |
|-----------------|---|-----------------------------|---|
| Входные данные | 3 4 1 2 3 4 5 6 7 8 9 10 11 12 | 2 3 10 20 30 40 50 60 | 5 5 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -21 -22 -23 -24 -25 |
| Вывод программы | (2, 0) | (1,0) | (0, 4) |

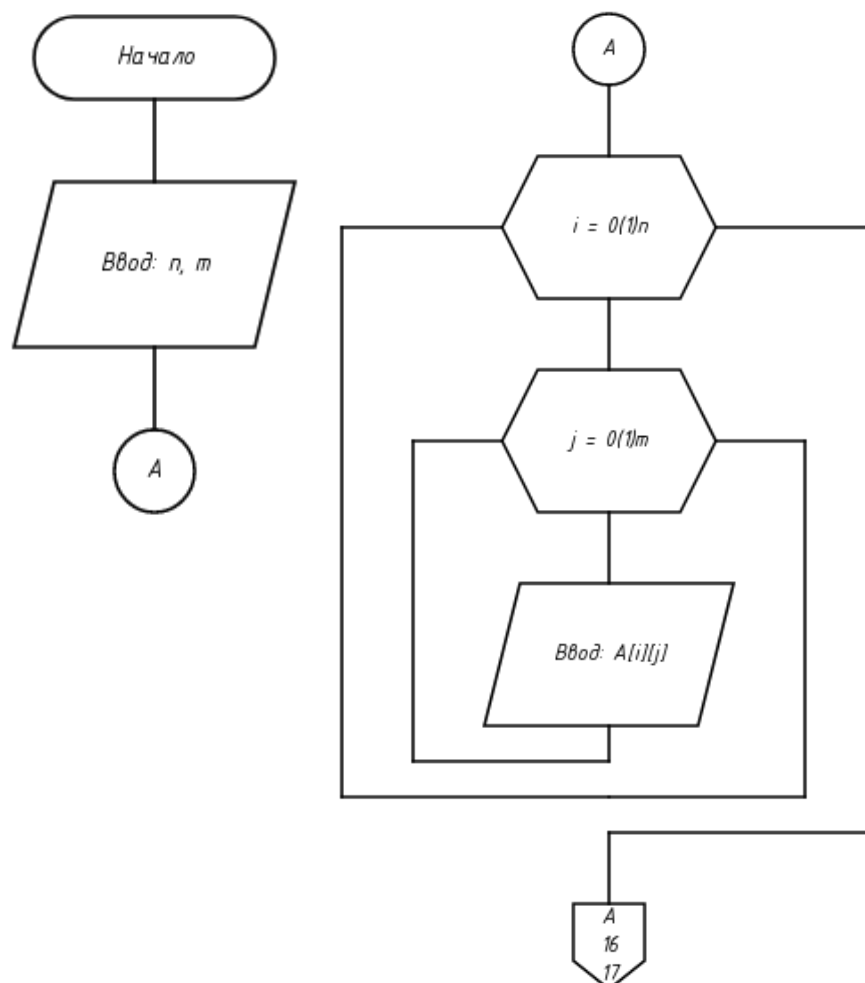
Задание 5:

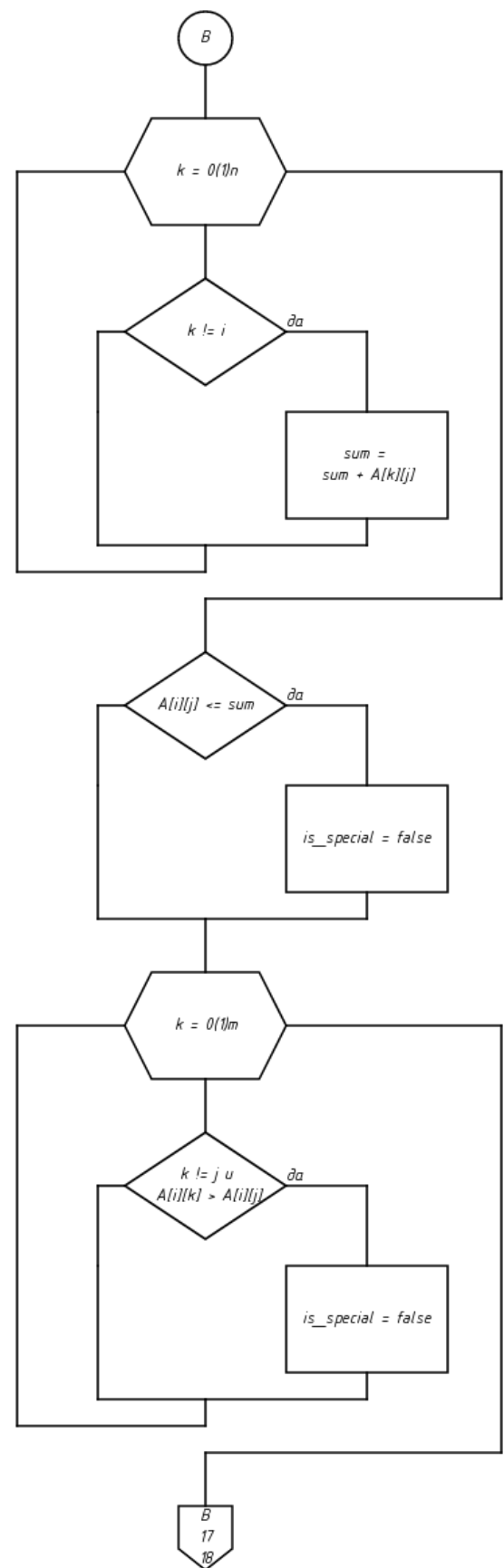
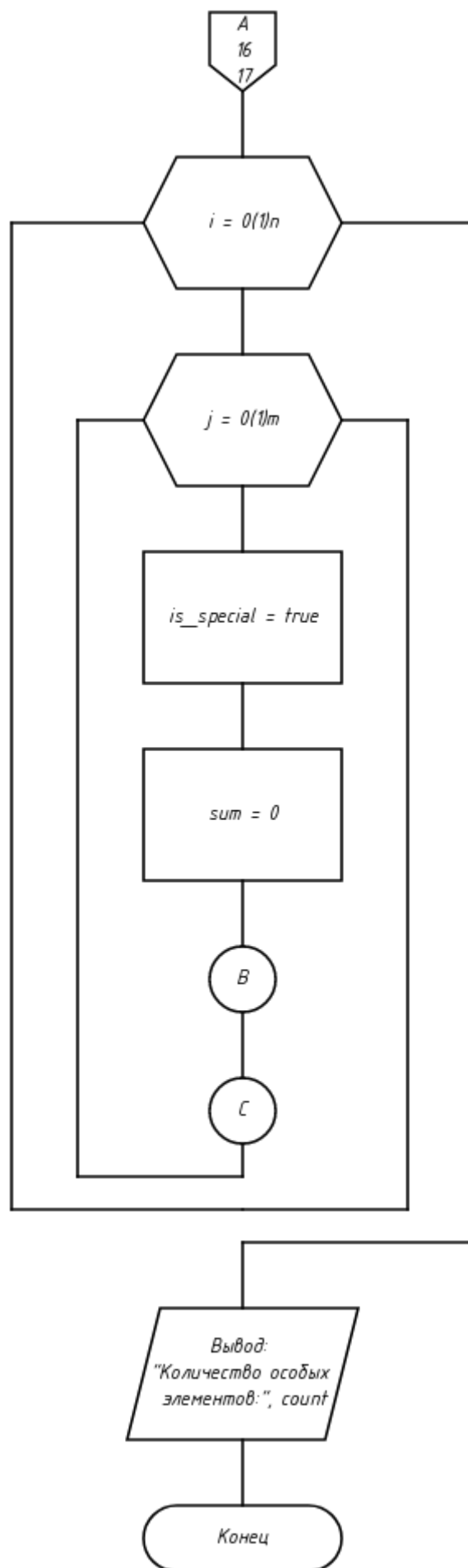
Формулировка:

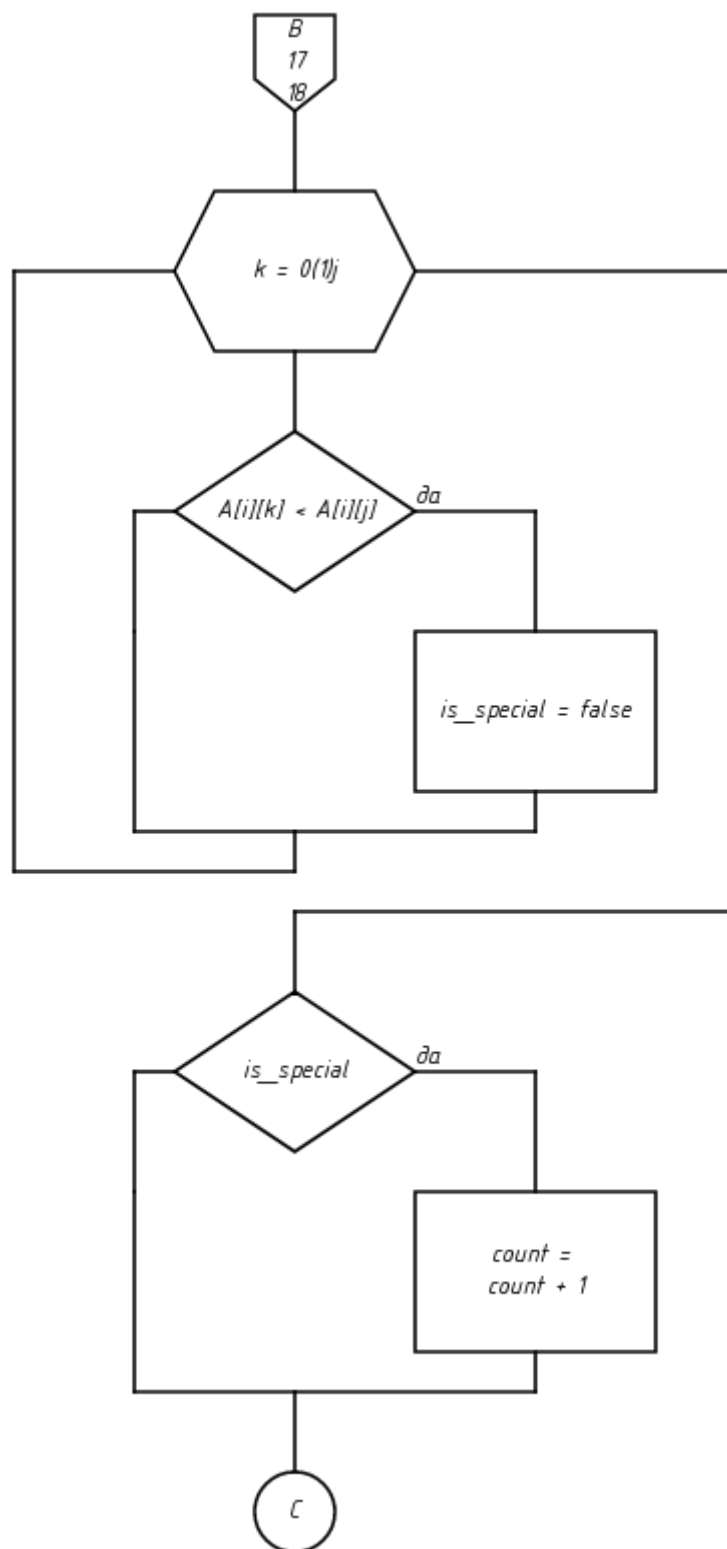
Дана матрица A (n , m). Определить количество особых элементов в ней, считая элемент особым, если:

- а) он больше суммы остальных элементов своего столбца;
- б) в его строке слева от него находятся элементы меньше его, а справа - больше

Блок схема:







Листинг программы на C++:

```

#include <iostream>
using namespace std;

int main()
{
  int n{}, m{};
  cin >> n >> m;

  int A[n][m];

```

```

for (int i = 0; i < n; i++)
{
for (int j = 0; j < m; j++)
{
cin >> A[i][j];
}
}

int count{}; // Счетчик особых элементов

// Проверяем каждый элемент матрицы
for (int i = 0; i < n; i++)
{
for (int j = 0; j < m; j++)
{
bool is_special = true;

// Проверка условия (а)
int sum = 0;
for (int k = 0; k < n; k++)
{
if (k != i)
{
sum += A[k][j];
}
}
if (A[i][j] <= sum)
{
is_special = false;
}

// Проверка условия (б)
for (int k = 0; k < m; k++)
{
if (k != j && A[i][k] > A[i][j])
{
is_special = false;
break;
}
}
for (int k = 0; k < j; k++)
{
if (A[i][k] < A[i][j])
{
is_special = false;
break;
}
}

if (is_special)
{
count++;
}
}
}

cout << "Количество особых элементов: " << count << endl;

return 0;
}

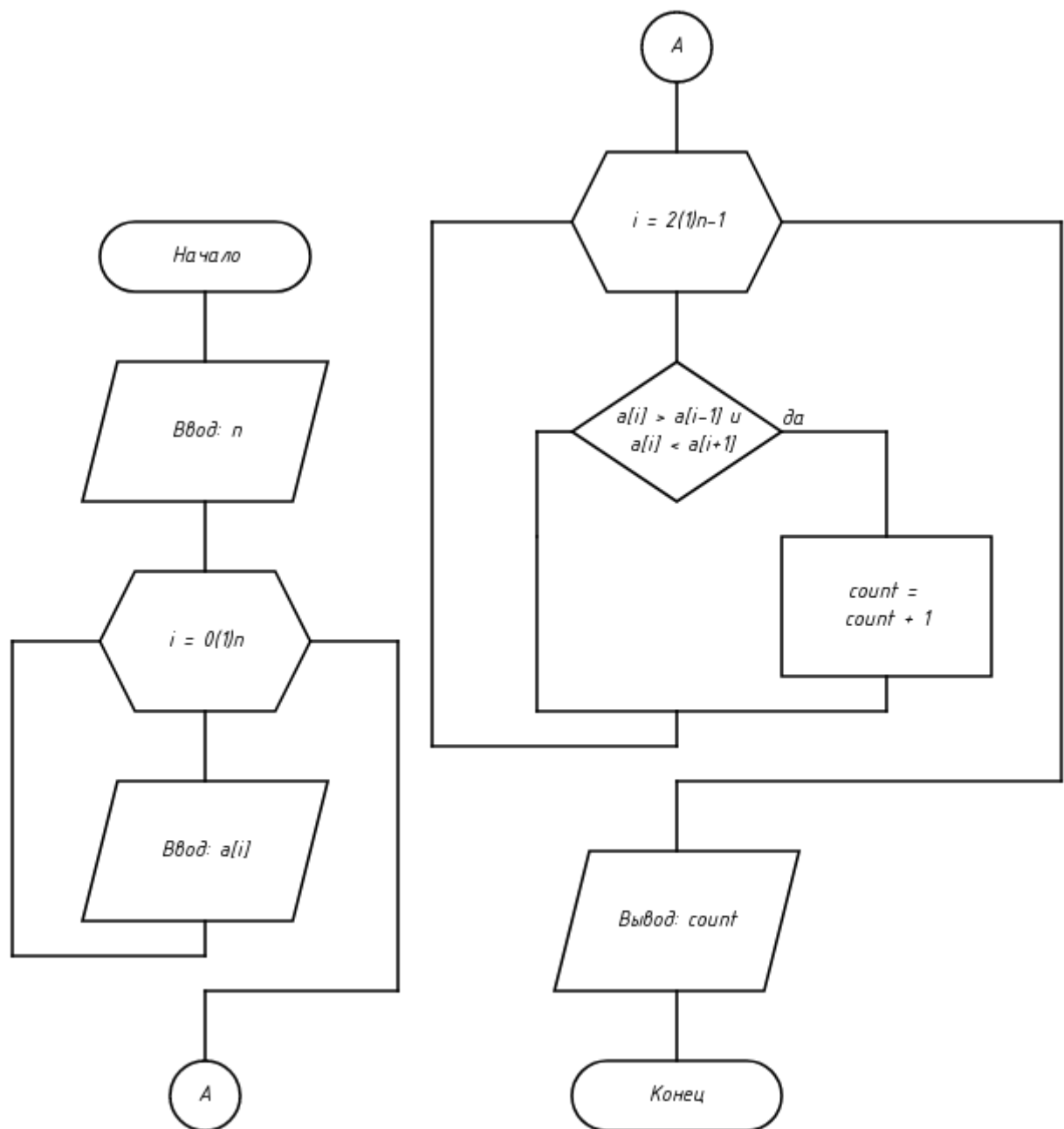
```

Задание 6:

Формулировка:

Даны: натуральное число n , действительные числа a_1, a_2, \dots, a_n .
Получить количество членов данной последовательности, которые больше предыдущего члена, но меньше последующего.

Блок схема:



Листинг программы на C++:

```
#include <iostream>
using namespace std;

int main() {
    int n{};
    cin >> n;

    double a[n];
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int count{};
    for (int i = 1; i < n - 1; i++) {
        if (a[i] > a[i-1] && a[i] < a[i+1]) {
            count++;
        }
    }

    cout << count << endl;

    return 0;
}
```

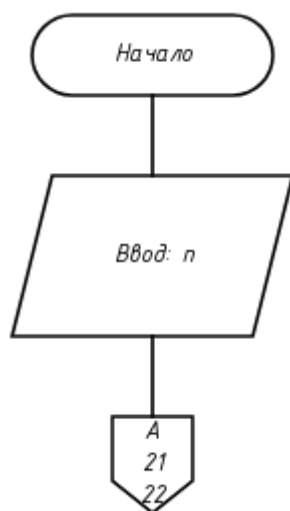
Задание 7:

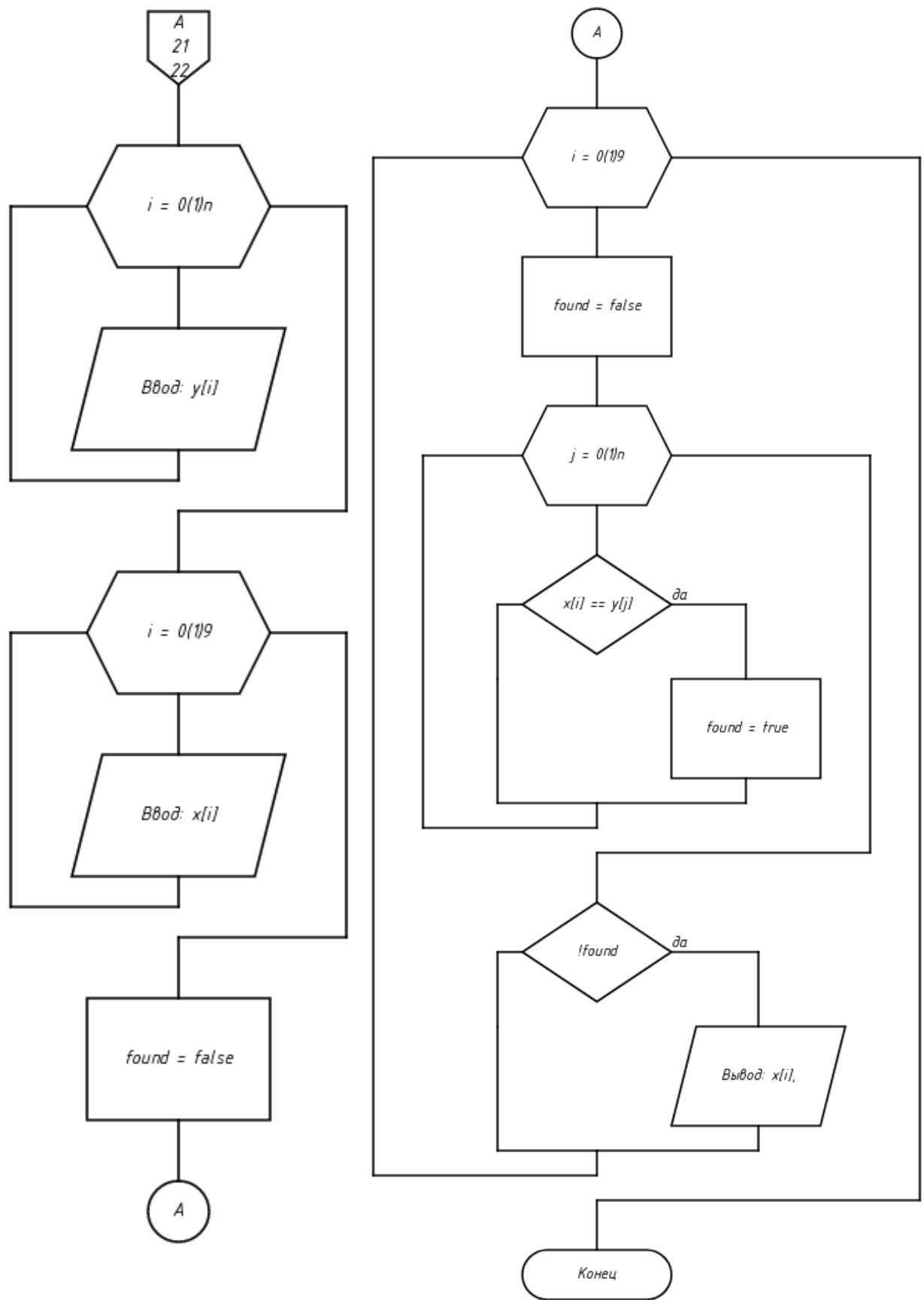
Формулировка:

Дано: натуральное число n , действительные числа x_1, x_2, \dots, x_9 и y_1, y_2, \dots, y_n .

Получить все члены последовательности x_1, x_2, \dots, x_9 , которые не входят в последовательность y_1, y_2, \dots, y_n .

Блок схема:





Листинг программы на C++:

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;

    double y[n];
    for (int i = 0; i < n; i++)
    {
        cin >> y[i];
    }

    double x[9]{};
    for (int i = 0; i < 9; i++)
    {
        cin >> x[i];
    }
    bool found = false;
    for (int i = 0; i < 9; i++)
    {
        found = false;
        for (int j = 0; j < n; j++)
        {
            if (x[i] == y[j])
            {
                found = true;
                break;
            }
        }
        if (!found)
        {
            cout << x[i] << ", ";
        }
    }
    return 0;
}
```

Вывод

В ходе данной лабораторной работы я изучил приемы работы с одномерными массивами средствами языка C++, разработал программы с использованием массивов, познакомился с операциями обработки элементов в массивах и изучил типовые алгоритмы решения задач с использованием массивов.

Литература

1. Курс лекций доцента кафедры ФН1-КФ Пчелинцевой Н.И.
2. Зырянов, К. И. Программирование на С++: учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Новосибирск: Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 с. — ISBN 978-5-7795-0817- 9. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/85873.html>
3. Фридман, А. Л. Язык программирования С++: учебное пособие / А. Л. Фридман. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. — ISBN 978-5-4497-0920-2. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102076.html>