

## 1. Цель работы

Целью работы является изучение структуры данных двумерного массива, научиться совершать операции над двумерным массивом.

## 2. Задание

Согласно варианту №1 требуется написать программу, с помощью которой можно заполнить целочисленную прямоугольную матрицу, определить количество строк, не содержащих ни одного нулевого элемента, вывести в консоль максимальное из чисел, встречающихся в заданной матрице более одного раза.

## 3. Описание созданных функций

Для реализации задания были использованы следующие функции:

**Имя:** Fillandoutput.

**Назначение:** заполнение и вывод массива.

**Входные данные:**

- arr[][4] – матрица размером 3x4;
- n – переменная, хранящая в себе размер строк матрицы;
- m – переменная, хранящая в себе количество столбцов матрицы.

**Выходные данные:**

- вывод матрицы в консоль .

**Побочный эффект:** отсутствует.

**Прототип:** Fillandoutput (int arr[][4],size\_t n, size\_t m);

**Алгоритм:**

- псевдокод

Ввод сообщения в консоль;

В цикле заполнения матрицы на каждой итерации цикла происходит рандомная генерация числа и его вывод в консоль, в следствии чего и строится матрица.

**Имя:** CheckZeroElem.

**Назначение:** проверка матрицы на наличие нулевых элементов.

**Входные данные:**

- arr[][4] – матрица размером 3x4;
- n – переменная, хранящая в себе размер строк матрицы;

- $m$  – переменная, хранящая в себе количество столбцов матрицы.

#### Выходные данные:

- count1- переменная- счётчик, хранящая в себе число строк содержащих нулевой элемент.

**Побочный эффект:** отсутствует.

**Прототип:** Fillandoutput (int arr[][4],size\_t n, size\_t m);

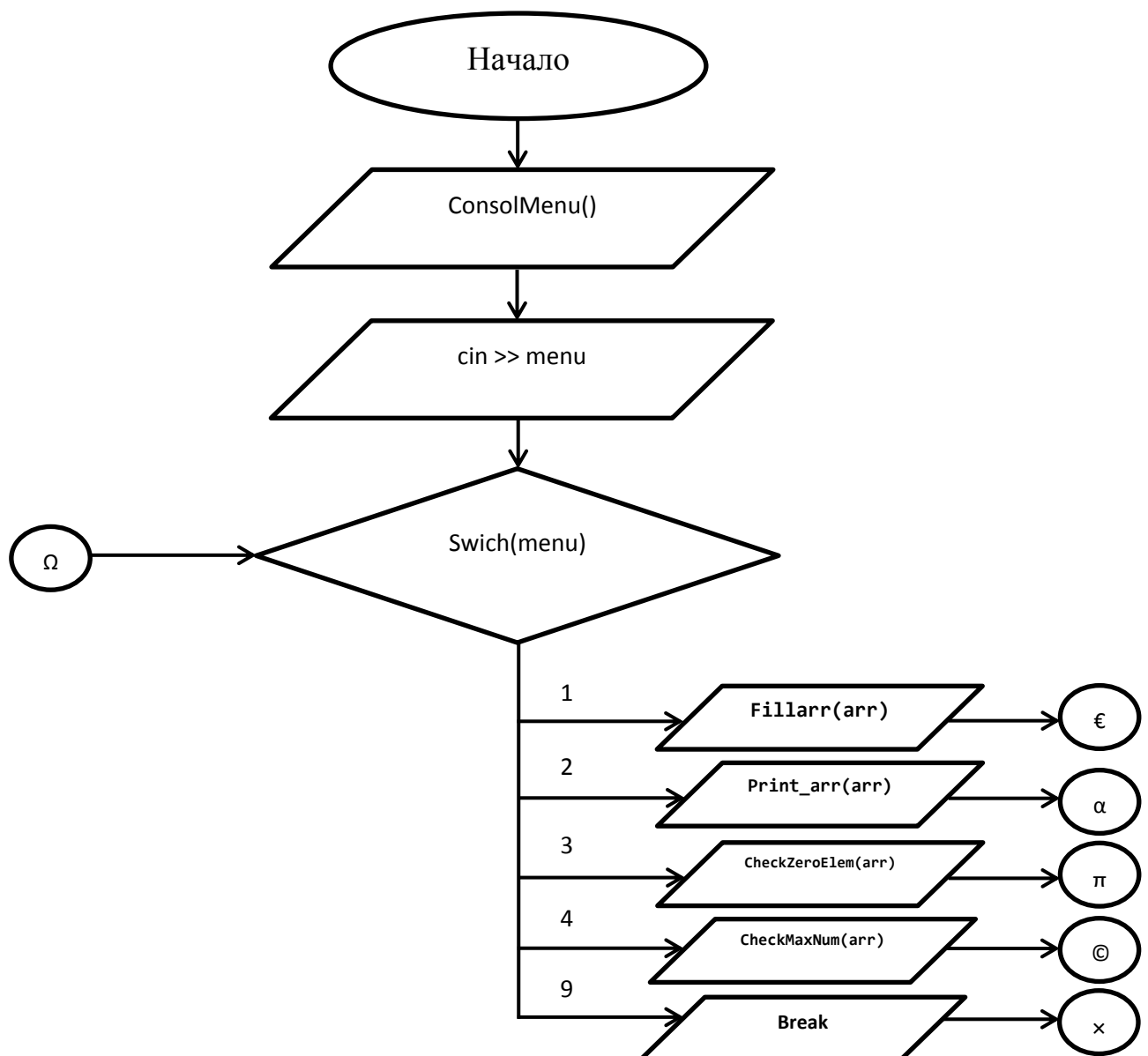
#### Алгоритм:

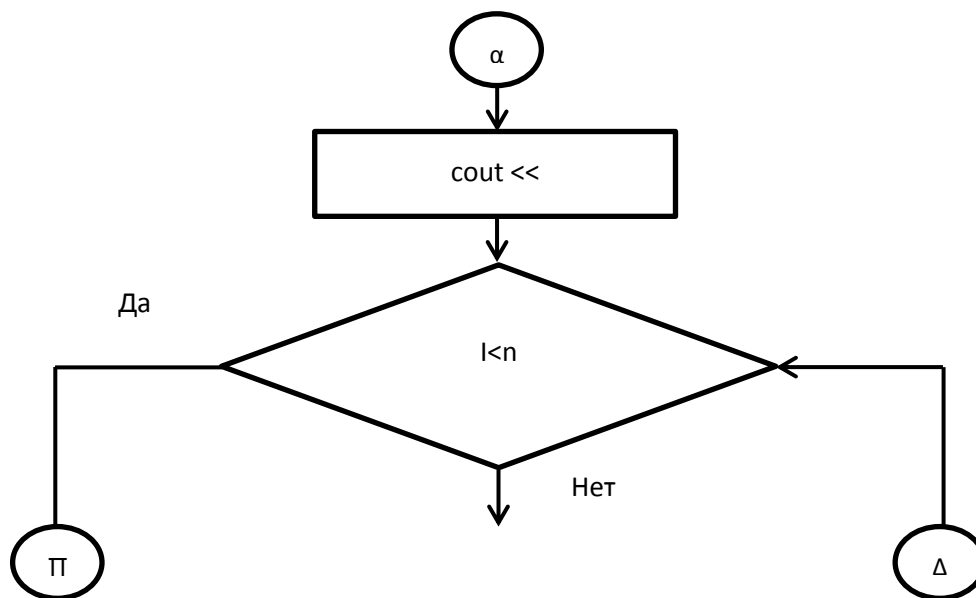
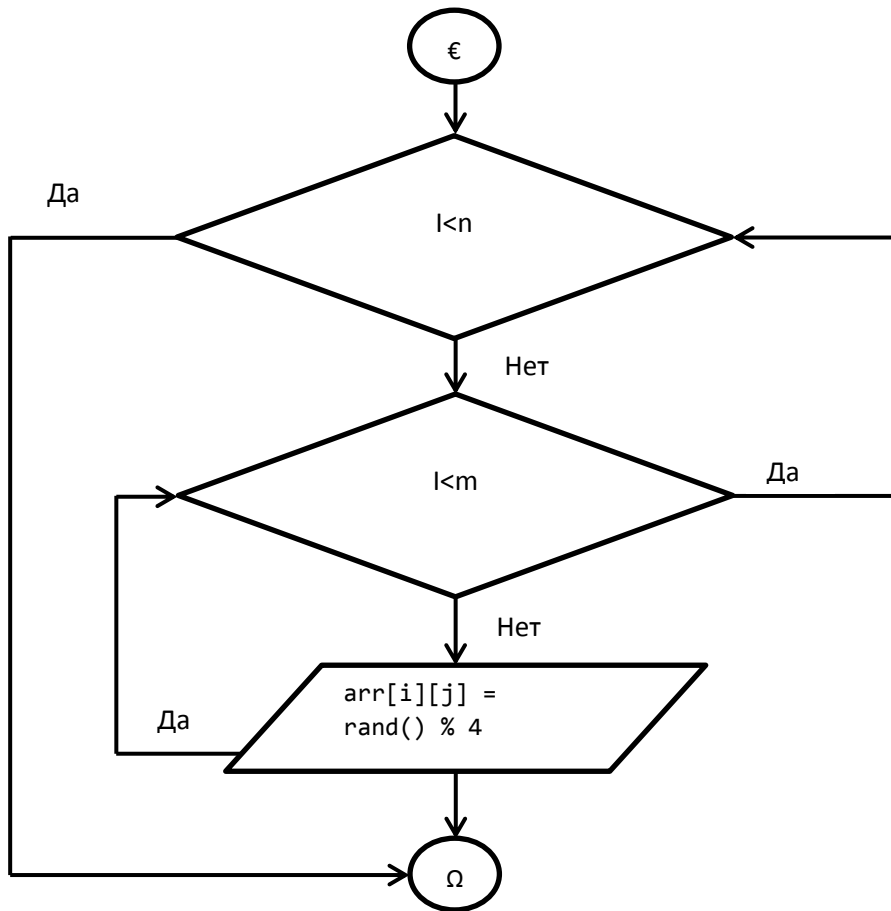
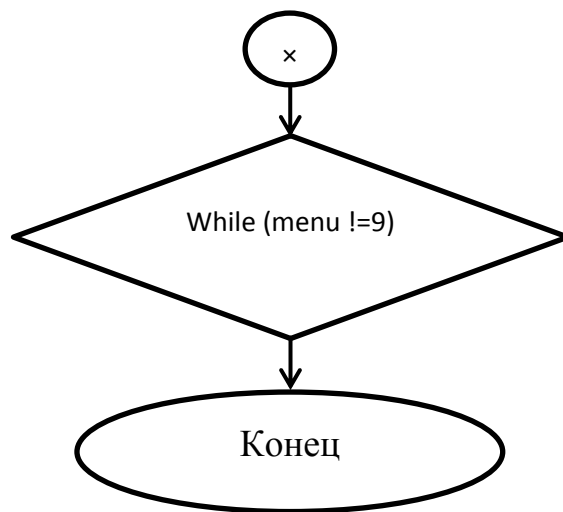
- псевдокод

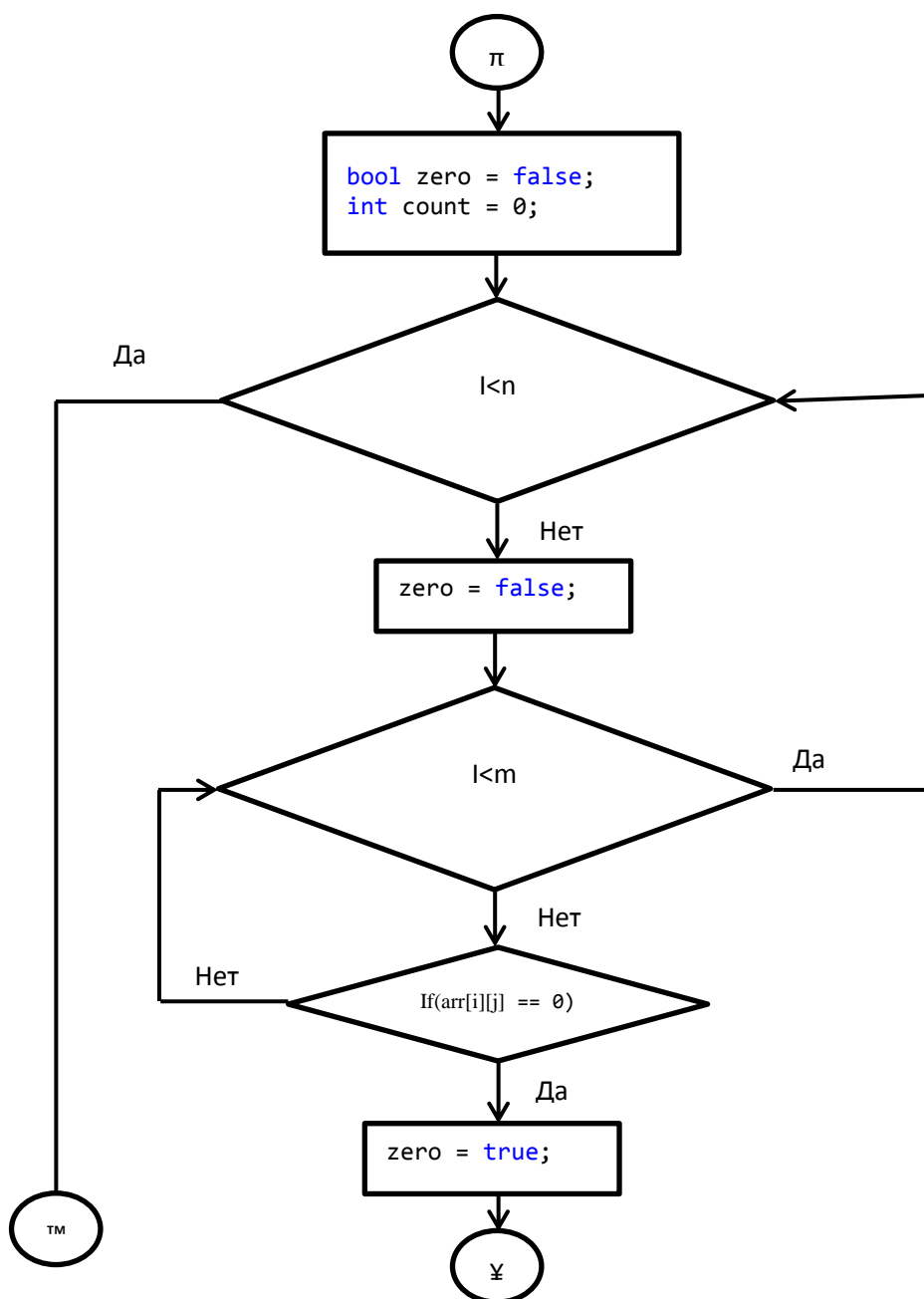
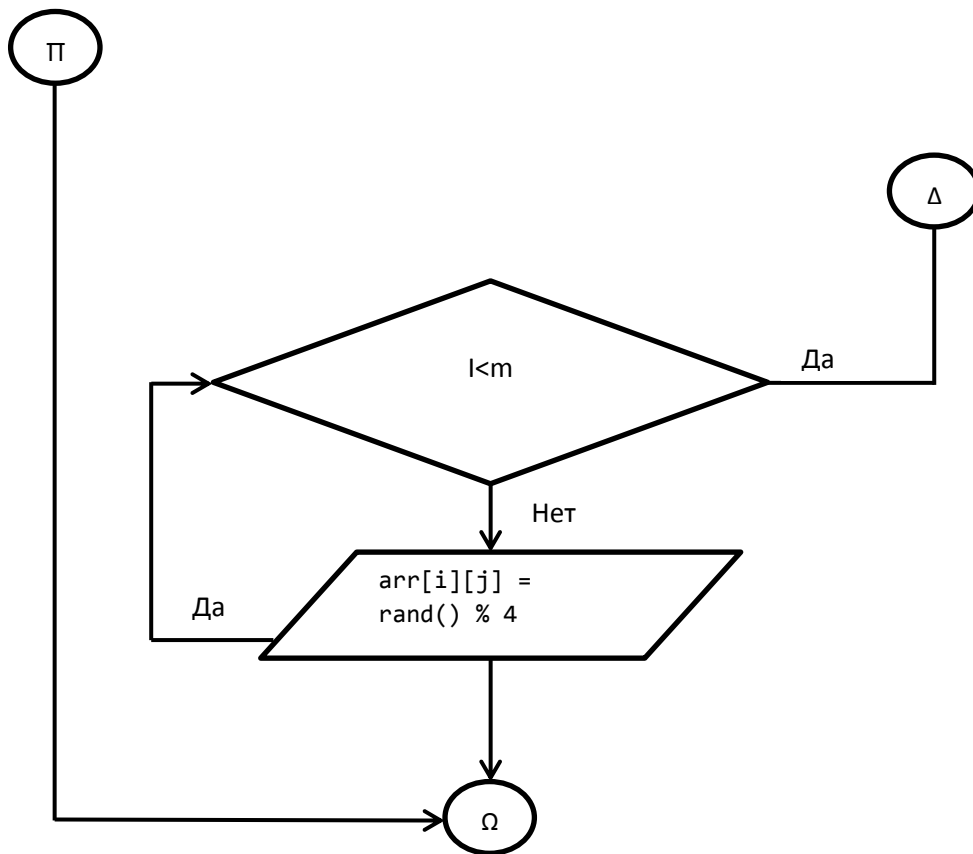
Ввод сообщения в консоль;

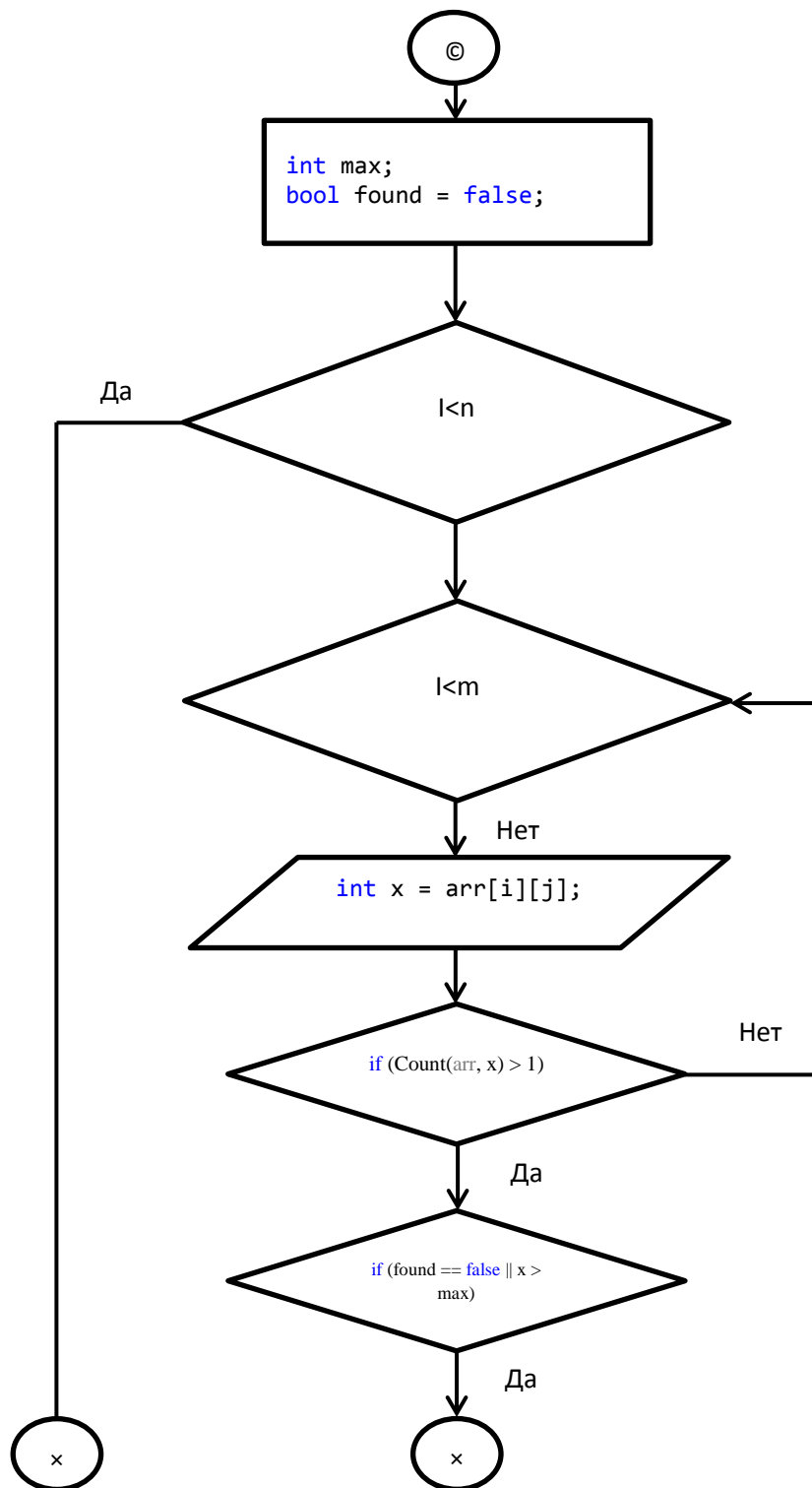
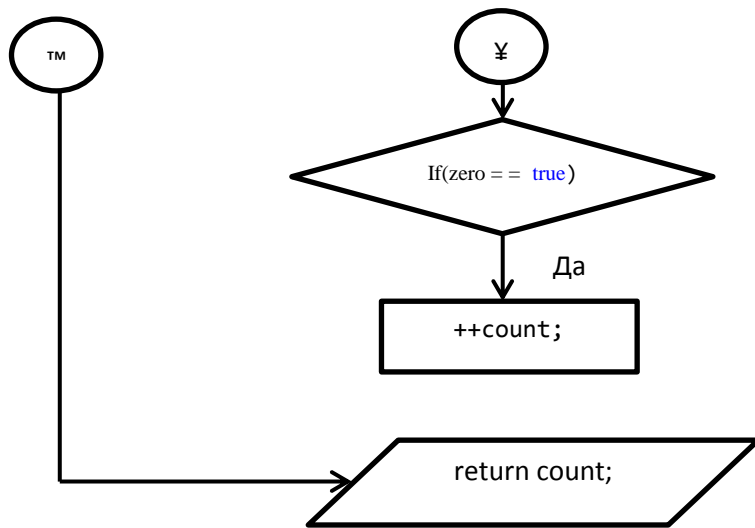
В цикле заполнения матрицы на каждой итерации цикла происходит рандомная генерация числа и его вывод в консоль, в следствии чего и строится матрица.

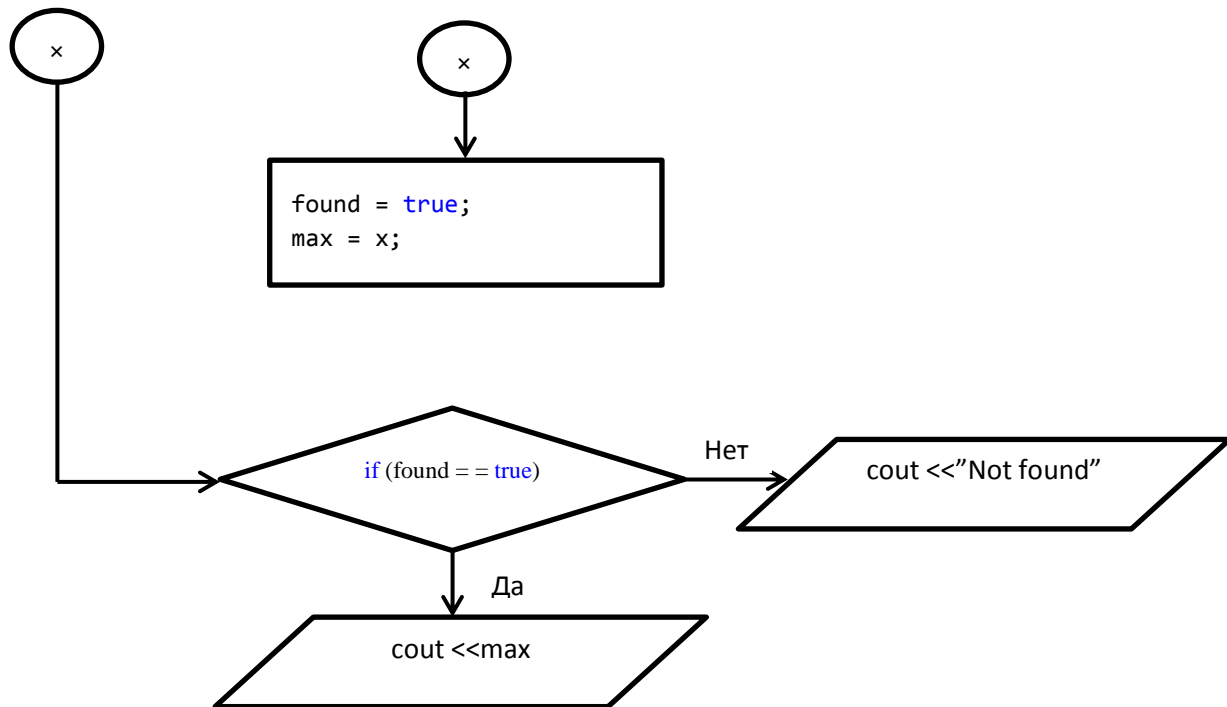
- Блок-схема











## 4. Листинг программы

---

```
1. #include <iostream>
2. #include <time.h>

3. using namespace std; // подключение имён пространства std

4. const int n = 3; // глобальная константная переменная для хранения числа строк
   матрицы
5. const int m = 4; // глобальная константная переменная для хранения числа столбцов
   матрицы

6. void Fillarr (int arr[n][m])
7. {
8.     for (int i(0); i < n; i++) // начало цикла с предусловием
9.     {
10.        for (int j(0); j < m; j++) // начало цикла с предусловием
11.        {
12.            arr[i][j] = rand() % 4; // заполнение матрицы случайными значениями от 0 до
               указанного значения после %
13.        }
14.    }
15. }

16. void Print_arr(int arr[n][m])
17. {
18.     cout << "\t\t\t\t\tМатрица 3x4\t\t\t\t\t\n\n"; // вывод в консоль

19.     for (int i(0); i < n; i++) // начало цикла с предусловием
20.     {
21.         cout << "|"; // вывод в консоль
22.         for (int j(0); j < m; j++) // начало цикла с предусловием
23.         {
24.             cout.width(10); // задаём пространство длиной в 10 пунктов для вывода
25.             cout << arr[i][j]; // выводим матрицу
26.             cout << "    |";
27.         }
28.         cout << "    " << endl;
29.     }
30.     cout << "\n";
31. }

32. int Count(int arr[n][m], int value) // функция для подсчёта повторений
33. {
34.     int c = 0; // переменная - счётчик
35.     for (int i = 0; i < n; i++)
36.     {
37.         for (int j = 0; j < m; j++)
38.         {
39.             if (arr[i][j] == value) // значения массива сравниваются с указанным значением
40.                 c++; // если находится повторяющееся значение, то счётчик инкрементируется
41.         }
42.     }
43.     return c; // возвращаем значение переменной-счётчика
44. }

45. void CheckMaxNum(int arr[n][m]) // поиск максимального значения , встречающегося в
   матрице более 2 раз
46. {
47.     int max;
48.     bool found = false;
```

```

49. for (int i = 0; i < n; i++)
50. {
51.     for (int j = 0; j < m; j++)
52.     {
53.         int x = arr[i][j]; // присваиваем переменной значение массива
54.         if (Count(arr, x) > 1) // условие при котором если заданное число x встречается
            больше 1 раза в матрице ,то
55.         {
56.             if (found == false || x > max)// если флаг false или x больше max
57.             {
58.                 found = true; // флаг true
59.                 max = x; // переменной max присвоить значение переменной x
60.             }
61.         }
62.     }
63. }

64. if (found == true)
65.     cout << "Максимальное число встречающееся больше одного раза: " << max << endl;
66. else
67.     cout << "Максимальное число встречающееся более одного раза не найдено" << endl;
68. }

69. int CheckZeroElem(int arr[n][m]) // начало функции проверки на наличие нулевых
    элементов
70. {
71.     bool zero = false; // переменная - флаг
72.     int count = 0; // переменная - счётчик
73.     for (int i = 0; i < n; i++) // объявление цикла
74.     {
75.         zero = false; // намеренное переопределение переменной-флага для каждого прохода
76.         for (int j = 0; j < m; j++) //объявление вложенного цикла
77.         {
78.             if (arr[i][j] == 0) // условие вложенного цикла, если элемент массива arr[i][j]
                равен 0, то флаг становится в положение true
79.             {
80.                 zero = true; // перевод переменной- флага в положение true
81.             }
82.         }
83.         if (zero == true) // условие основного цикла
84.         {
85.             ++count; // инкрементирование переменной- счётчика
86.         }
87.     }
88.     return count; // возвращаем значение переменной-счётчика после всех вычислений
89. }

90. void ConsolMenu() // функция для инициализации меню
91. {
92.     cout << "1. Заполнить матрицу рандомногенерируемыми числами\n";// вывод в консоль
93.     cout << "2. Вывод матрицы\n";
94.     cout << "3. Проверить матрицу на нулевые элементы\n";
95.     cout << "4. Проверить матрицу количество максимально повторяющихся элементов более
        1 раза";
96.     cout << "9. Выход" << endl;
97. }

98. int main() // основная функция
99. {
100.     setlocale(LC_ALL, "Russian"); // использование русского языка
101.     srand(time(NULL));
102.     int menu, zero = 0; //
103.     int arr[n][m]; // объявление переменных

104.     do

```



```
105.     {
106.     ConsolMenu();
107.     cin >> menu;
108.     cout << "\n";
109.     switch (menu)
110.     {
111.     case 1: Fillarr(arr); cout << "\n"; system("cls"); break;
112.     case 2: Print_arr(arr); break;
113.     case 3:
114.     {
115.     cout << "\n";
116.     cout << "Кол-во строк содержащие 0 : " << CheckZeroElem(arr) << "\n\n";
117.     break;
118.     }
119.     case 4:
120.     {
121.     CheckMaxNum(arr)// поиск максимального значения , встречающегося в матрице
        более 2 раз
122.     ; cout << "\n"; break;
123.     }
124.     case 9: system("cls"); break;
125.     }
126.     } while (menu != 9);

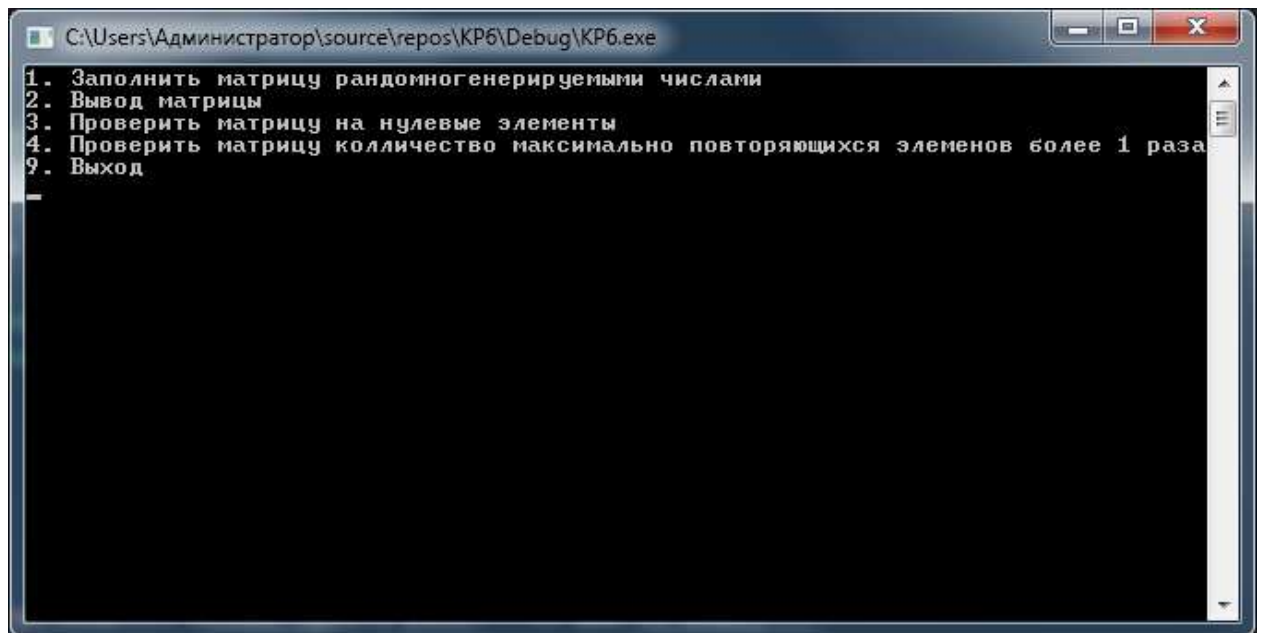
127.     cout << "\n\n"; // вывод в консоль

128.     system("pause"); // подключение системной команды пауза
129.     return 0; // возвращение значений
130. }
```

---

## 5. Пример выполнения программы

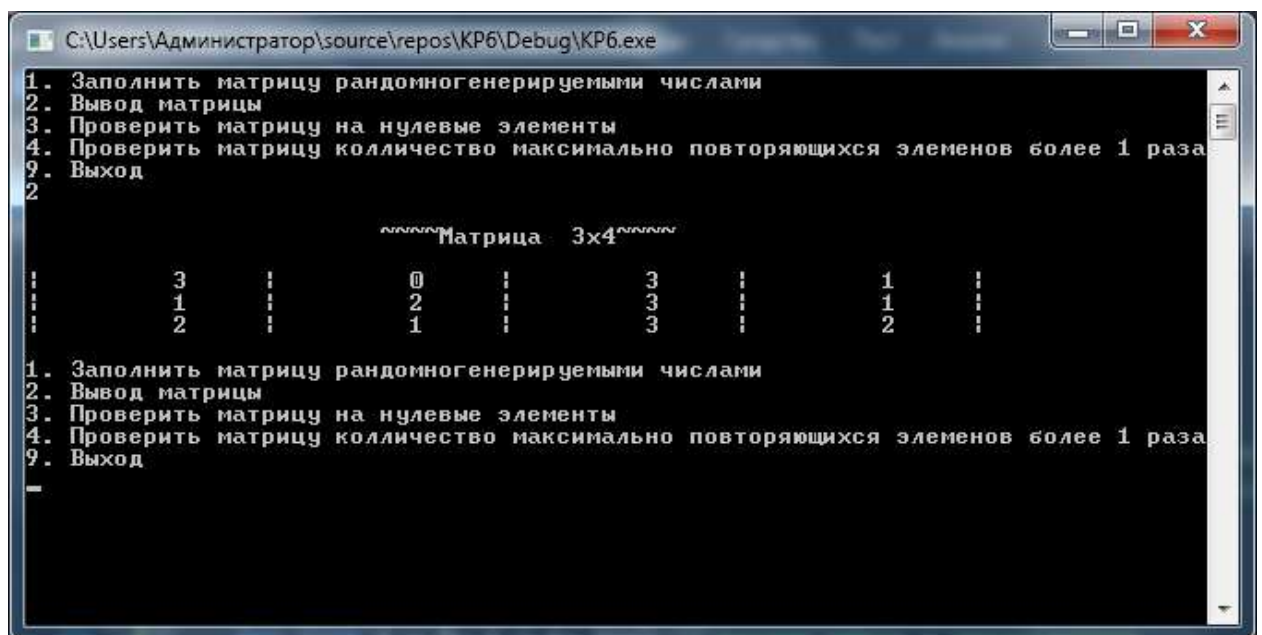
Ниже приведён пример выполнения программы



```
C:\Users\Администратор\source\repos\KP6\Debug\KP6.exe

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
```

Рис 1. Пример выполнения программы



```
C:\Users\Администратор\source\repos\KP6\Debug\KP6.exe

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
2

~~~~~Матрица 3x4~~~~~
:      3      :      0      :      3      :      1      :
:      1      :      2      :      3      :      1      :
:      2      :      1      :      3      :      2      :

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
-
```

Рис 2. Пример выполнения программы

```
C:\Users\Администратор\source\repos\KP6\Debug\KP6.exe

9. Выход
2

~~~~~Матрица 3x4~~~~~
:      3      :      0      :      3      :      1      :
:      1      :      2      :      3      :      1      :
:      2      :      1      :      3      :      2      :

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
3

Кол-во строк содержащие 0 :1

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
```

Рис 3. Пример выполнения программы

```
C:\Users\Администратор\source\repos\KP6\Debug\KP6.exe

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
3

Кол-во строк содержащие 0 :1

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
4

Максимальное число встречающееся больше одного раза: 3

1. Заполнить матрицу рандомногенерируемыми числами
2. Вывод матрицы
3. Проверить матрицу на нулевые элементы
4. Проверить матрицу количество максимально повторяющихся элементов более 1 раза
9. Выход
```

Рис 4. Пример выполнения программы