

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное  
учреждение высшего образования  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
РУТ (МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**ОТЧЕТ ПО ДИСЦИПЛИНЕ  
«АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»  
ЛАБОРАТОРНАЯ РАБОТА №7**

*Направление: 10.03.01 Информационная безопасность*

*Профиль: Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-115  
Шевченко Димитрий

Проверил:

---

(должность, ФИО)

---

(должность, ФИО)

Москва 2021 г.

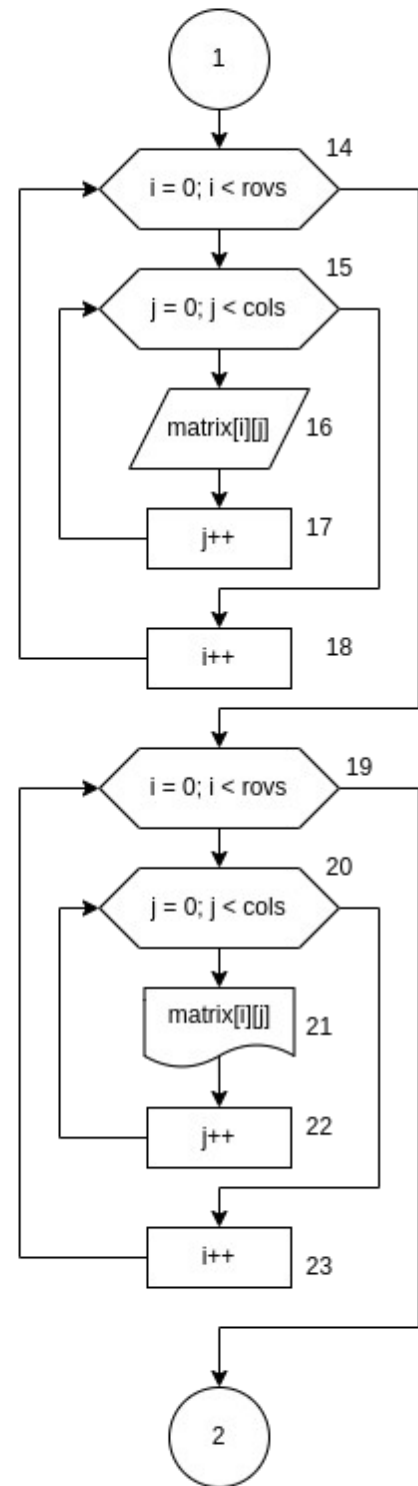
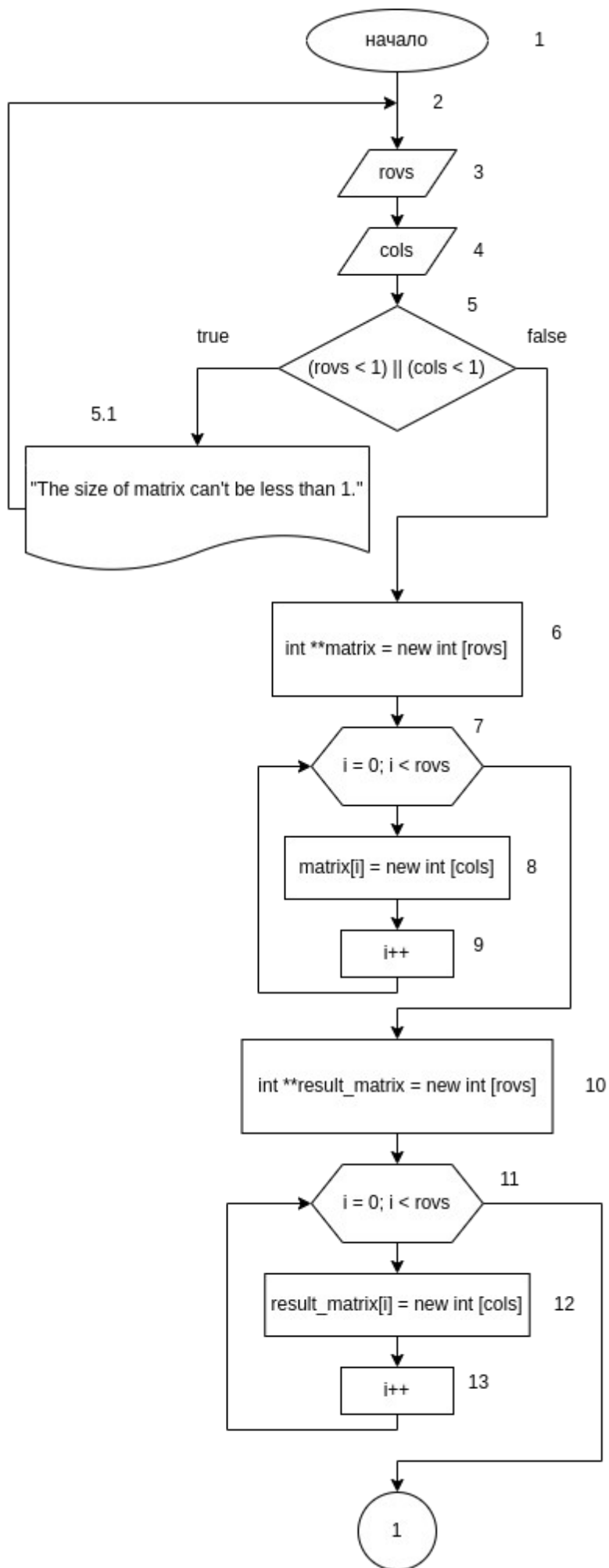
## Задание №7

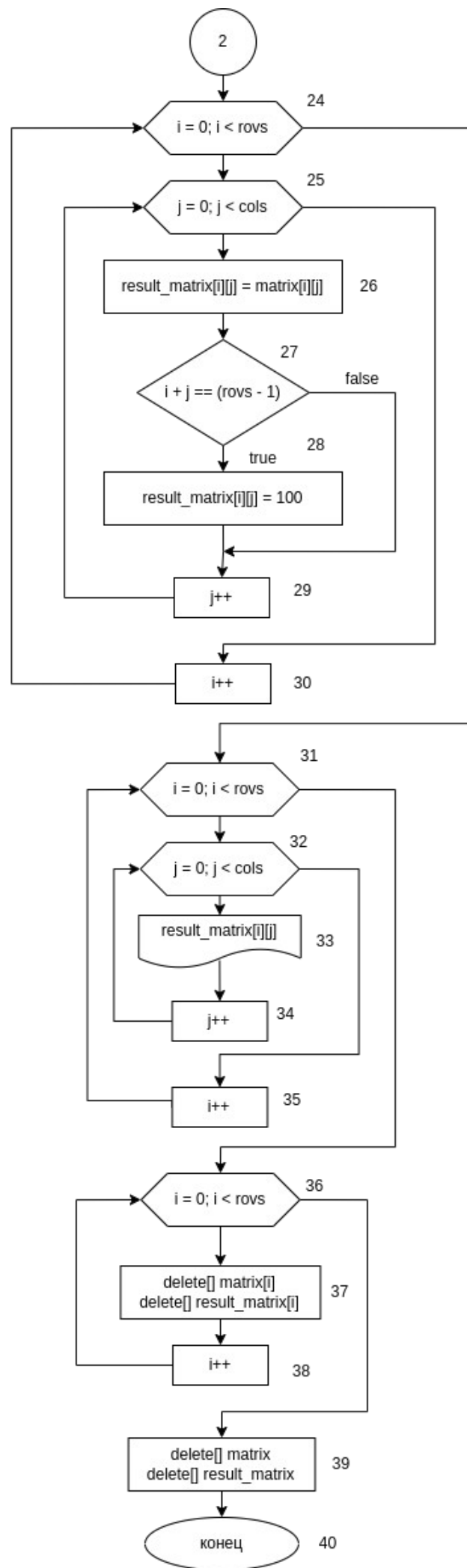
Задание: Задана матрица целых чисел. Заменить все значения элементов побочной диагонали матрицы значениями, равными 100.

### 1 Таблица имён:

Исходные данные		
rows	Целочисленное (int)	Количество строк
cols	Целочисленное (int)	Количество столбцов
matrix[][]	Целочисленное (int)	Исходная матрица
Рабочие переменные		
i	Целочисленное (int)	Счётчик
j	Целочисленное (int)	Счётчик
Результат		
result_matrix[][]	Целочисленное (int)	Изменённая матрица

### 2 Блок-схема:





### 3 Отладочные примеры:

#### Вариант I

1) Начало

2) Бесконечный цикл

3) Ввод rows

5

4) Ввод cols

5

5) Так как 5 больше 1, то останавливаем цикл

6) Создаём массив matrix с размером 5

7) Цикл от 0 до 5 с переменной i

8) matrix[i] = new int [5]

9) i++

10) Создаём массив result\_matrix с размером 5

11) Цикл от 0 до 5 с переменной i

12) result\_matrix[i] = new int [5]

13) i++

16) Ввод matrix[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

21) Вывод matrix[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

24) Цикл от 0 до 5 с переменной i

25) Цикл от 0 до 5 с переменной j

26) result\_matrix[i][j] = matrix[i][j]

27) Если (i+j) равны (rovs-1), то

28) result\_matrix[i][j] = 100

29) j++

30) i++

33) Вывод result\_matrix[5][5]

1	2	3	4	100
6	7	8	100	10
11	12	100	14	15
16	100	18	19	20
100	22	23	24	25

36) Цикл от 0 до 5 с переменной i

37) Удаление из памяти matrix[i] и result\_matrix[i]

38) i++

39) Удаление из памяти matrix и result\_matrix

40) Конец

## Вариант II

1) Начало

2) Бесконечный цикл

3) Ввод rows

-3

4) Ввод cols

0

5) Так как -3 и 0 меньше 1, то возвращаемся к шагу 3

3) Ввод rows

3

4) Ввод cols

2

5) Так как 3 и 2 больше 1, то останавливаем цикл

6) Создаём массив matrix с размером 3

7) Цикл от 0 до 3 с переменной i

8) matrix[i] = new int [2]

9) i++

10) Создаём массив result\_matrix с размером 3

11) Цикл от 0 до 3 с переменной i

12) result\_matrix[i] = new int [2]

13) i++

16) Ввод matrix[3][2]

9

8

7

6

5

4

21) Вывод matrix[3][2]

9    8

7    6

5    4

24) Цикл от 0 до 3 с переменной i  
25) Цикл от 0 до 2 с переменной j  
26) result\_matrix[i][j] = matrix[i][j]  
27) Если (i+j) равны (rovs-1), то  
28) result\_matrix[i][j] = 100  
29) j++  
30) i++

33) Вывод result\_matrix[3][2]

9     8  
7     100  
100   4

36) Цикл от 0 до 3 с переменной i  
37) Удаление из памяти matrix[i] и result\_matrix[i]  
38) i++

39) Удаление из памяти matrix и result\_matrix  
40) Конец



### Вариант III

1) Начало

2) Бесконечный цикл

3) Ввод rows

3

4) Ввод cols

6

5) Так как 3 и 6 больше 1, то останавливаем цикл

6) Создаём массив matrix с размером 3

7) Цикл от 0 до 3 с переменной i

8) matrix[i] = new int [6]

9) i++

10) Создаём массив result\_matrix с размером

11) Цикл от 0 до 3 с переменной i

12) result\_matrix[i] = new int [6]

13) i++

16) Ввод matrix[3][6]

132

654

798

654

987

546

312

012

904

456

753

951

852

963

741

486

426

682

21) Вывод matrix[3][6]

132	654	798	654	987	546
312	12	904	456	753	951
852	963	741	486	426	682

24) Цикл от 0 до 3 с переменной i

25) Цикл от 0 до 6 с переменной j

26) result\_matrix[i][j] = matrix[i][j]

27) Если (i+j) равны (rovs-1), то

28) result\_matrix[i][j] = 100

29) j++

30) i++

33) Вывод result\_matrix[3][6]

132	654	100	654	987	546
312	100	904	456	753	951
100	963	741	486	426	682

36) Цикл от 0 до 3 с переменной i

37) Удаление из памяти matrix[i] и result\_matrix[i]

38) i++

39) Удаление из памяти matrix и result\_matrix

40) Конец

#### 4 Код программы:

```
#include <iostream> //подключение библиотеки функции ввода-вывода
using namespace std; //подключение пространства имён std

int main(){ //точка входа в программу

    int rovs, cols;
    // ввод размера матрицы пока размер не будет больше 0
    while (true){
        cout << "Input rovs of matrix: " << endl;
        cin >> rovs;

        cout << "Input cols of matrix: " << endl;
        cin >> cols;

        if ((rovs < 1) || (cols < 1)){
            cout << "The size of matrix can't be less than 1." << endl;
            cout << "-----" << endl;
        }
        else {
            cout << "-----" << endl;
            break;
        }
    }
    // Создание исходной матрицы
    int **matrix = new int *[rovs];
    for (int i = 0; i < rovs; i++){
        matrix[i] = new int [cols];
    }
    // Создание матрицы результата
    int **result_matrix = new int *[rovs];
    for (int i = 0; i < rovs; i++){
        result_matrix[i] = new int [cols];
    }
    // Ввод исходной матрицы
    cout << "Input matrix's elements: " << endl;
    for (int i = 0; i < rovs; i++){
        for (int j = 0; j < cols; j++){
            cin >> matrix[i][j];
        }
    }

    cout << "-----" << endl;
```

```

for (int i = 0; i < rovs; i++){ // Вывод исходной матрицы
    for (int j = 0; j < cols; j++){
        cout << matrix[i][j] << "\t";
    }
    cout << endl;
}

cout << "-----" << endl;

// заполнение изменённой матрицы
for (int i = 0; i < rovs; i++){
    for (int j = 0; j < cols; j++){
        result_matrix[i][j] = matrix[i][j]; // передача значений к новой
        матрице
        if (i + j == (rovs — 1)){ // Если сумма i и j равна количеству
        строк, уменьшенных на 1, то
            result_matrix[i][j] = 100; // Присваиваем текущему
            элементу значение 100
        }
    }
}

// вывод изменённой матрицы
for (int i = 0; i < rovs; i++){
    for (int j = 0; j < cols; j++){
        cout << result_matrix[i][j] << "\t";
    }
    cout << endl;
}

// Очистка ОЗУ от указателей
for (int i = 0; i < rovs; i++){
    delete[] matrix[i];
    delete[] result_matrix[i];
}

delete[] matrix;
delete[] result_matrix;

return 0; //успешное завершение программы
}

```

5 Результат выполнения работы программы:

1)

```
Input rovs of matrix:
5
Input cols of matrix:
5
-----
Input matrix's elements:
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
-----
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
-----
1      2      3      4      100
6      7      8      100    10
11     12     100    14     15
16     100    18     19     20
100    22     23     24     25
```

2)

```
Input rovs of matrix:
-3
Input cols of matrix:
0
The size of matrix can't be less than 1.
-----
Input rovs of matrix:
3
Input cols of matrix:
2
-----
Input matrix's elements:
9
8
7
6
5
4
-----
9      8
7      6
5      4
-----
9      8
7      100
100    4
```

3)

```
Input rovs of matrix:
3
Input cols of matrix:
6
-----
Input matrix's elements:
132
654
798
654
987
546
312
012
904
456
753
951
852
963
741
486
426
682
-----
132      654      798      654      987      546
312      12      904      456      753      951
852      963      741      486      426      682
-----
132      654      100      654      987      546
312      100      904      456      753      951
100      963      741      486      426      682
```

## 6 Вывод:

В ходе выполнения работы были изучены алгоритмы поиска побочной и главной диагоналей матрицы.

Была проделана работа по созданию UI, алгоритма поиска и замены значений побочной диагонали матрицы.

На контрольных примерах мы убедились, что программа работает корректно и отвечает заданным в ТЗ требованиям.

Был оформлен комплект документации на программный код.