

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 №8**

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.

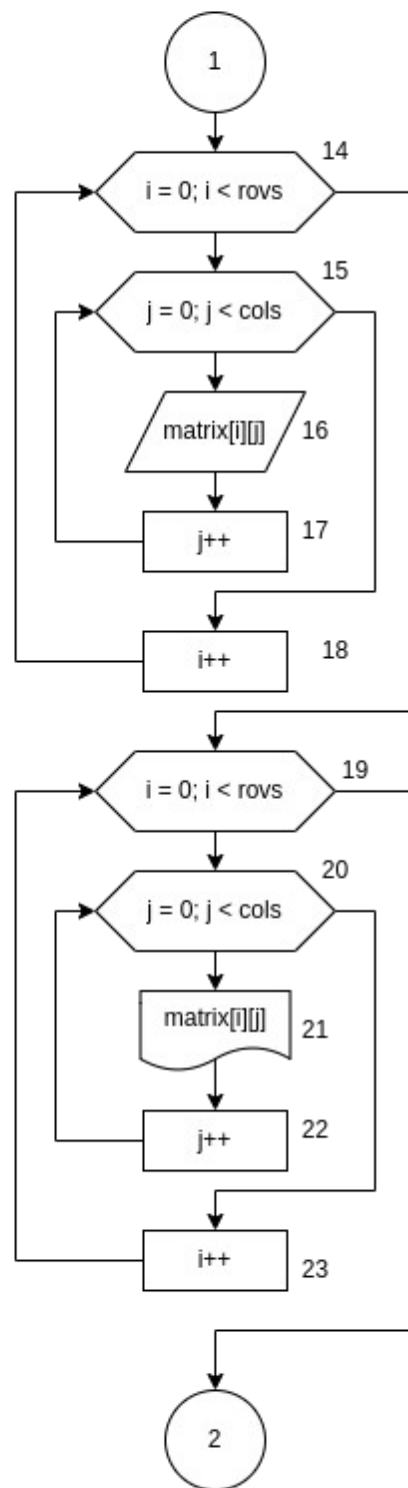
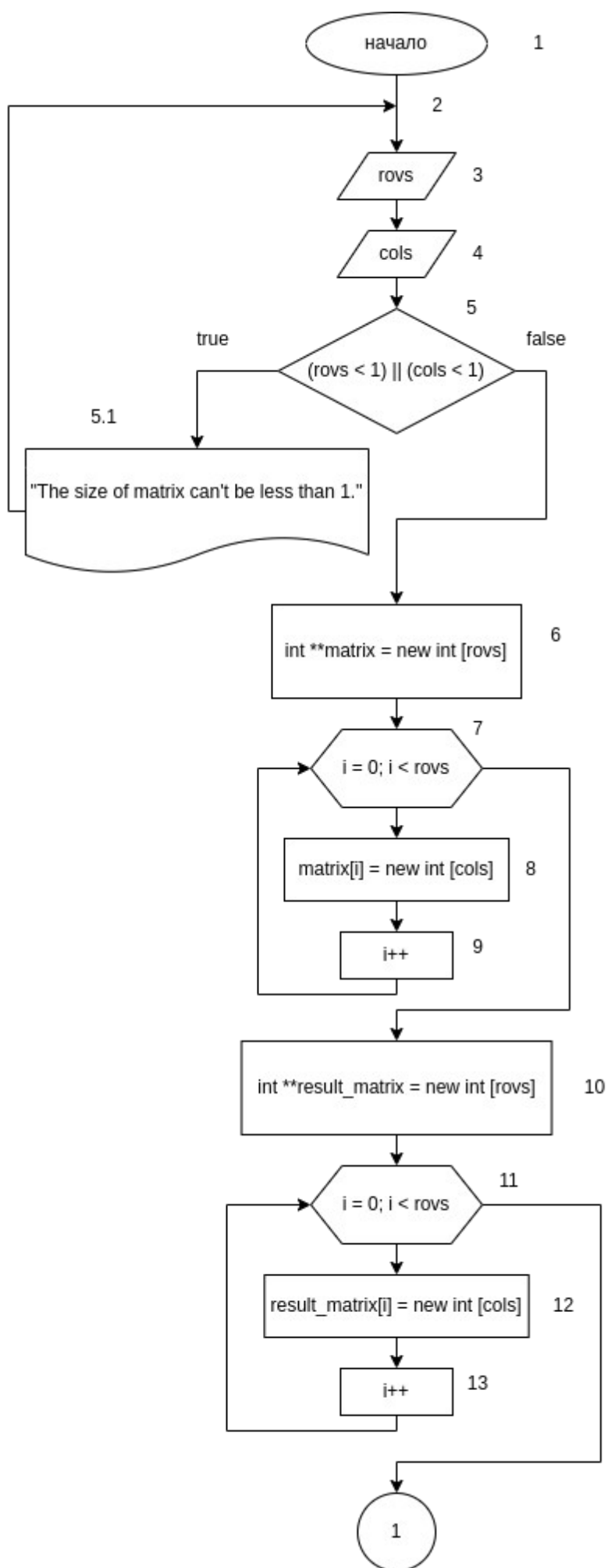
Задание №8

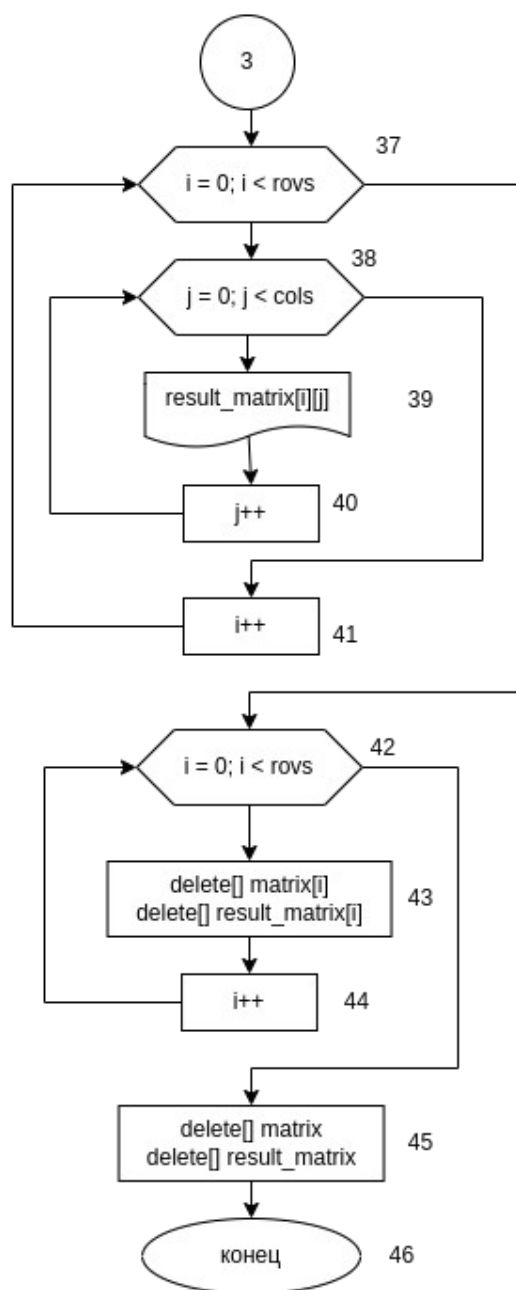
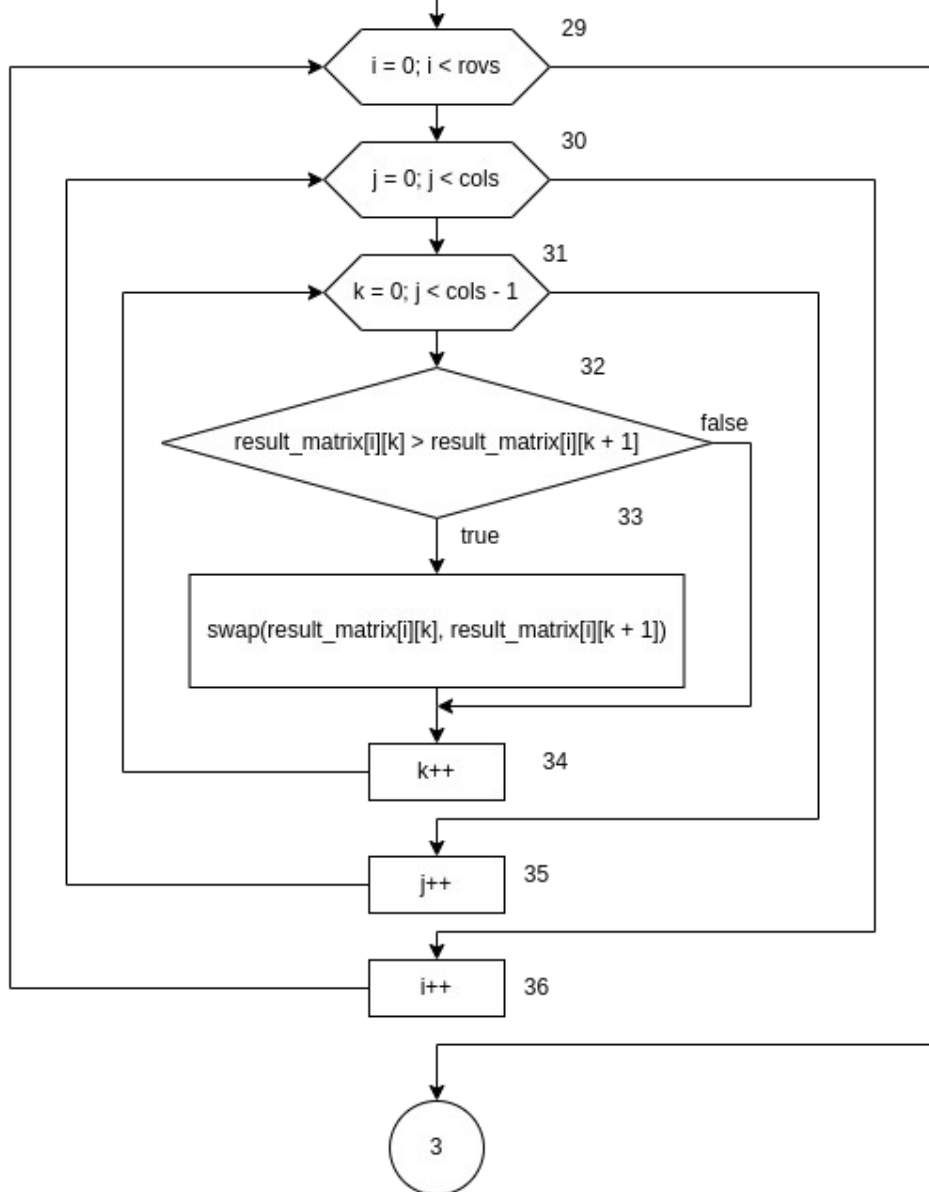
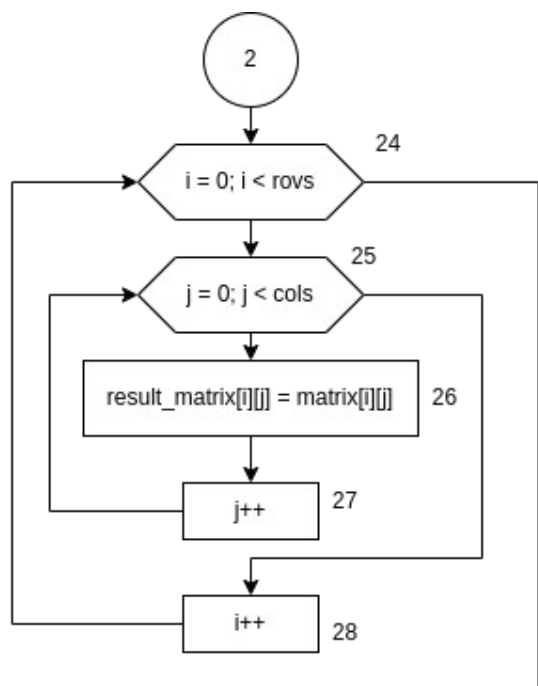
Задание: Задана матрица целых чисел. Выполнить сортировку элементов в каждой строке матрицы по возрастанию.

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

3

4) Ввод cols

3

5) Так как 3 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 [3]

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 [3]

13) i++

16) Ввод matrix[3][3]

9

7

8

7

1

4

9

4

6

21) Вывод matrix[3][3]

9 7 8

7 1 4

9 4 6

- 24) Цикл от 0 до 3 с переменной i
 - 25) Цикл от 0 до 3 с переменной j
 - 26) result_matrix[i][j] = matrix[i][j]
 - 27) j++
 - 28) i++
- 29) Цикл от 0 до 3 с переменной i
 - 30) Цикл от 0 до 3 с переменной j
 - 31) Цикл от 0 до 2 с переменной k
 - 32) Если result_matrix[i][k] больше result_matrix[i][k + 1], то
 - 33) Меняем result_matrix[i][k] и result_matrix[i][k + 1] местами
 - 34) k++
 - 35) j++
 - 36) i++

39) Вывод result_matrix[3][3]

7	8	9
1	4	7
4	6	9

- 42) Цикл от 0 до 3 с переменной i
 - 43) Удаление из памяти matrix[i] и result_matrix[i]
 - 44) i++

45) Удаление из памяти matrix и result_matrix

46) Конец

Вариант II

1) Начало

2) Бесконечный цикл

3) Ввод rows

-5

4) Ввод cols

-2

5) Так как -5 и -2 меньше 1, то возвращаемся к шагу 3

3) Ввод rows

5

4) Ввод cols

2

5) Так как 5 и 2 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 [2]

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 [2]

13) i++

16) Ввод matrix[5][2]

9

8

7

6

5

4

3

2

1

0

21) Вывод matrix[5][2]

9	8
7	6
5	4
3	2
1	0

24) Цикл от 0 до 5 с переменной i

25) Цикл от 0 до 2 с переменной j

26) result_matrix[i][j] = matrix[i][j]

27) j++

28) i++

29) Цикл от 0 до 5 с переменной i

30) Цикл от 0 до 2 с переменной j

31) Цикл от 0 до 1 с переменной k

32) Если result_matrix[i][k] больше result_matrix[i][k + 1], то

33) Меняем result_matrix[i][k] и result_matrix[i][k + 1]
местами

34) k++

35) j++

36) i++

39) Вывод result_matrix[5][2]

8	9
6	7
4	5
2	3
0	1

42) Цикл от 0 до 5 с переменной i

43) Удаление из памяти matrix[i] и result_matrix[i]

44) i++

45) Удаление из памяти matrix и result_matrix

46) Конец

Вариант III

1) Начало

2) Бесконечный цикл

3) Ввод rows

2

4) Ввод cols

5

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

6) Создаём массив matrix с размером 2

7) Цикл от 0 до 2 с переменной i

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

9) i++

10) Создаём массив result_matrix с размером 2

11) Цикл от 0 до 2 с переменной i

12) result_matrix[i] = new int [5]

13) i++

16) Ввод matrix[2][5]

89

61

-15

-18

-70

83

49

-35

81

46

21) Вывод matrix[2][5]

89 61 -15 -18 -70

83 49 -35 81 46

24) Цикл от 0 до 2 с переменной i

25) Цикл от 0 до 5 с переменной j

26) result_matrix[i][j] = matrix[i][j]

27) j++

28) i++

29) Цикл от 0 до 2 с переменной i

30) Цикл от 0 до 5 с переменной j

31) Цикл от 0 до 4 с переменной k

32) Если result_matrix[i][k] больше result_matrix[i][k + 1], то

33) Меняем result_matrix[i][k] и result_matrix[i][k + 1]
местами

34) k++

35) j++

36) i++

39) Вывод result_matrix[2][5]

-70 -18 -15 61 89

-35 46 49 81 83

42) Цикл от 0 до 2 с переменной i

43) Удаление из памяти matrix[i] и result_matrix[i]

44) i++

45) Удаление из памяти matrix и result_matrix

46) 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;

// передача значений в матрицу результата
for (int i = 0; i < rovs; i++){
    for (int j = 0; j < cols; j++){
        result_matrix[i][j] = matrix[i][j];
    }
}

// сортировка матрицы по возрастанию
for (int i = 0; i < rovs; i++){
    for (int j = 0; j < cols; j++){
        for (int k = 0; k < cols - 1; k++){
            if (result_matrix[i][k] > result_matrix[i][k + 1]){
                swap(result_matrix[i][k], result_matrix[i][k + 1]);
            }
        }
    }
}

// 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:
```

```
3
```

```
Input cols of matrix:
```

```
3
```

```
-----
```

```
Input matrix's elements:
```

```
9
```

```
7
```

```
8
```

```
7
```

```
1
```

```
4
```

```
9
```

```
4
```

```
6
```

```
-----
```

```
9
```

```
7
```

```
8
```

```
7
```

```
1
```

```
4
```

```
9
```

```
4
```

```
6
```

```
-----
```

```
7
```

```
8
```

```
9
```

```
1
```

```
4
```

```
7
```

```
4
```

```
6
```

```
9
```

2)

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

7
6

5
4

3
2

1
0

-----
9      8
7      6
5      4
3      2
1      0
-----
8      9
6      7
4      5
2      3
0      1
```

3)

```
Input rovs of matrix:
2
Input cols of matrix:
5
-----
Input matrix's elements:
89
61
-15
-18
-70

83
49
-35
81
46

-----
89      61      -15      -18      -70
83      49      -35      81      46
-----
-70      -18      -15      61      89
-35      46      49      81      83
```

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.