

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное
учреждение высшего образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Инженерная школа информационных технологий и робототехники
Отделение автоматизации и робототехники
Направление мехатроника и робототехника

Отчет
по лабораторной работе №3

по дисциплине
«Основы программирования и алгоритмизации»

Матрицы в C++

Выполнил:

Студент группы 8E21

Н.С. Моисеев

Проверил:

Ассистент ОАР ИШИТР

Я.О. Кургинов

Томск 2023

Вариант 10

Цель работы:

Поработать с матрицами в C++.

Задание 1

Дана матрица размера $M \times N$. Вывести ее элементы, расположенные в столбцах с нечетными номерами (1, 3, ...). Вывод элементов производить по столбцам, условный оператор не использовать.

Блок схема (Рисунок 1)

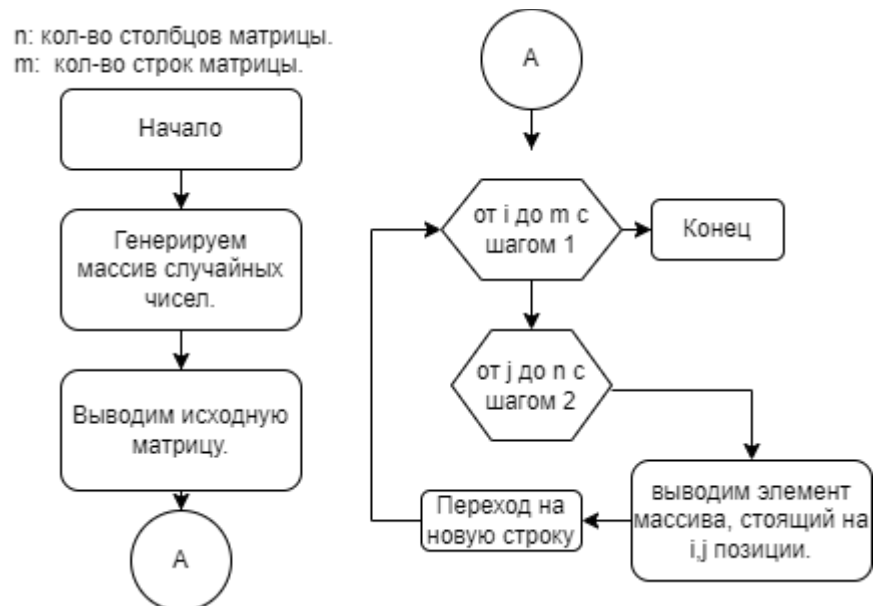


Рисунок 1 - Блок-схема к заданию 1.

Текст программы (Листинг 1, Приложение А)

Результаты работы (Рисунок 2)

```
Введите кол-во столбцов матрицы: 5
Введите кол-во строк матрицы: 3
Исходная матрица:
83 86 77 15 93
35 86 92 49 21
62 27 90 59 63

Нечётные столбцы матрицы:
83 77 93
35 92 21
62 90 63
```

Рисунок 2 - Скриншот результата работы программы к заданию 1 на консоли.

Задание 2:

Дана квадратная матрица порядка M . Обнулить элементы матрицы, лежащие выше побочной диагонали. Условный оператор не использовать.

Блок-схема (Рисунок 3)

n : порядок матрицы
 a : эл-т матрицы, стоящий на позиции $[i,j]$

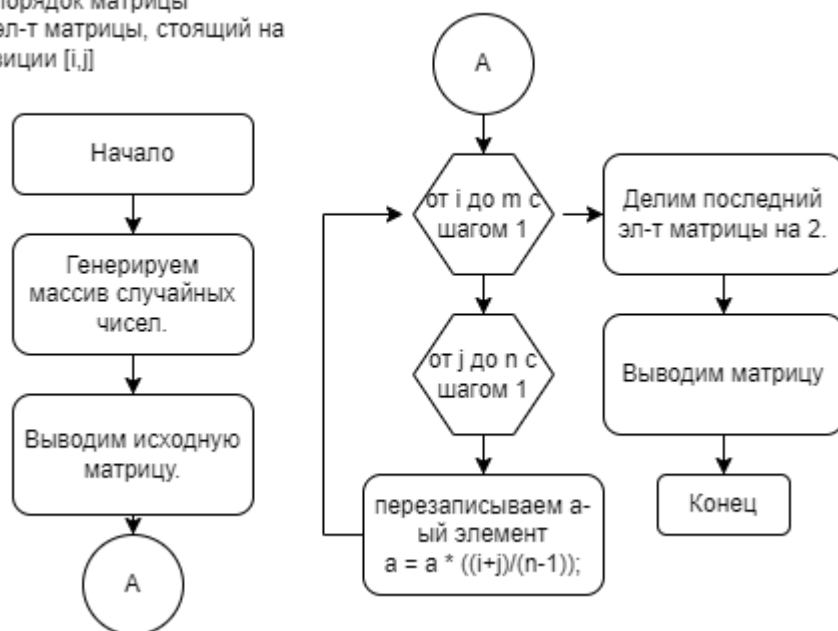


Рисунок 3 - Блок-схема к заданию 2.

Текст программы (Листинг 2, Приложение А)

Результаты работы (Рисунок 4)

```

Введите порядок матрицы: 3
Исходная матрица:
3 6 7
5 3 5
6 2 9

Обнуление элементов матрицы выше побочной диагонали матрицы:
0 0 7
0 3 5
6 2 9
    
```

Рисунок 4 - Скриншот результата работы программы к заданию 2 на консоли.

Задание 3

Дана матрица размера $M \times N$. Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждом столбце.

Блок схема (Рисунок 5)

n: кол-во столбцов матрицы.
 m: кол-во строк матрицы.
 a: эл-т матрицы, стоящий на позиции [i,j]

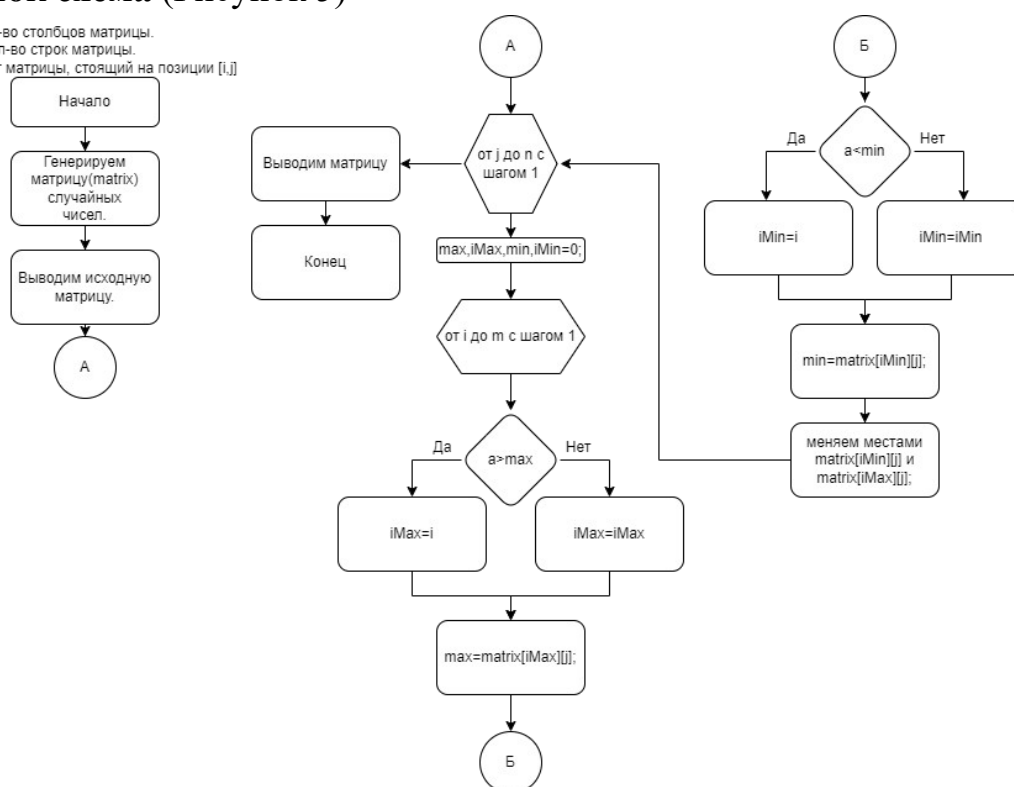


Рисунок 5 - Блок-схема к заданию 3.

Текст программы (Листинг 3, Приложение А)

Результаты работы (Рисунок 6)

```
Введите кол-во столбцов матрицы: 5
Введите кол-во строк матрицы: 3
Исходная матрица:
83 86 77 15 93
35 86 92 49 21
62 27 90 59 63

Отредактированная матрица:
35 27 92 59 21
83 86 77 49 93
62 86 90 15 63
```

Рисунок 6 - Скриншот результата работы программы к заданию 3 на консоли.

Выводы:

В результате лабораторной работы были решены поставленные задания с матрицами, используя C++.

Приложение А

Листинг 1 – Программа к заданию 1.

```
1. #include <iostream>
2.
3. int main()
4. {
5.     std::cout<<"Введите кол-во столбцов матрицы: "; int n; std::cin>>n;
6.     std::cout<<"Введите кол-во строк матрицы: "; int m; std::cin>>m;
7.     int** matrix = new int*[m];
8.     for (int i = 0; i < m; i++) {
9.         matrix[i] = new int[n];
10.    }
11.    for(int i=0;i<m;i++){
12.        for(int j=0;j<n;j++){
13.            matrix[i][j] = rand()%100;
14.        }
15.    }
16.    std::cout << std::endl;
17.
18.
19.    std::cout<<"Исходная матрица: " << std::endl;
20.    for(int i=0;i<m;i++){
21.        for(int j=0;j<n;j++){
22.            std::cout << matrix[i][j] << " ";
23.        }
24.        std::cout << std::endl;
25.    }
26.    std::cout << std::endl;
27.
```

```

28.
29.     std::cout<<"Нечётные столбцы матрицы: " << std::endl;
30.     for(int i=0;i<m;i++){
31.         for(int j=0;j<n;j+=2){
32.             std::cout << matrix[i][j] << " ";
33.         }
34.         std::cout << std::endl;
35.     }
36.
37.     for (int i = 0; i < m; i++)
38.         delete[] matrix[i];
39.     delete[] matrix;
40.
41.     return 0;
42. }

```

Листинг 2 – Программа к заданию 2.

```

1. #include <iostream>
2.
3. int main()
4. {
5.     std::cout<<"Введите порядок матрицы: "; int n; std::cin>>n;
6.     int** matrix = new int*[n];
7.     for (int i = 0; i < n; i++) {
8.         matrix[i] = new int[n];
9.     }
10.    for(int i=0;i<n;i++){
11.        for(int j=0;j<n;j++){
12.            matrix[i][j] = rand()%100;

```

```

13.         }
14.     }
15.     std::cout << std::endl;
16.
17.
18.     std::cout<<"Исходная матрица: " << std::endl;
19.     for(int i=0;i<n;i++){
20.         for(int j=0;j<n;j++){
21.             std::cout << matrix[i][j] << " ";
22.         }
23.         std::cout << std::endl;
24.     }
25.     std::cout << std::endl;
26.
27.
28.     std::cout<<"Обнуление элементов матрицы выше побочной диагонали матрицы: " <<
std::endl;
29.     for(int i=0;i<n;i++){
30.         for(int j=0;j<n;j++){
31.             matrix[i][j] = matrix[i][j] * ((i+j)/(n-1));
32.         }
33.     }
34.     matrix[n-1][n-1] = matrix[n-1][n-1]/2;
35.
36.
37.     for(int i=0;i<n;i++){
38.         for(int j=0;j<n;j++){
39.             std::cout << matrix[i][j] << " ";
40.         }
41.         std::cout << std::endl;

```



```
42.     }  
43.  
44.  
45.     for (int i = 0; i < n; i++)  
46.         delete[] matrix[i];  
47.     delete[] matrix;  
48.     return 0;  
49. }
```

Листинг 3 – Программа к заданию 3.

```
1. #include <iostream>  
2.  
3. int main()  
4. {  
5.     std::cout<<"Введите кол-во столбцов матрицы: "; int n; std::cin>>n;  
6.     std::cout<<"Введите кол-во строк матрицы:   "; int m; std::cin>>m;  
7.     int** matrix = new int*[m];  
8.     for (int i = 0; i < m; i++) {  
9.         matrix[i] = new int[n];  
10.    }  
11.    for(int i=0;i<m;i++){  
12.        for(int j=0;j<n;j++){  
13.            matrix[i][j]=rand()%100;  
14.        }  
15.    }  
16.  
17.  
18.    std::cout<<"Исходная матрица: " << std::endl;  
19.    for(int i=0;i<m;i++){
```

```

20.         for(int j=0;j<n;j++){
21.             std::cout << matrix[i][j] << " ";
22.         }
23.         std::cout << std::endl;
24.     }
25.     std::cout << std::endl;
26.
27.
28.     for(int j=0;j<n;j++){
29.         int max = 0;
30.         int iMax = 0;
31.         int min = 0;
32.         int iMin = 0;
33.         for(int i=0;i<m;i++){
34.             iMax=(matrix[i][j]>max)?i:iMax;
35.             max=matrix[iMax][j];
36.             iMin=(matrix[i][j]<min)?i:iMin;
37.             min=matrix[iMin][j];
38.         }
39.         int temp=matrix[iMin][j];
40.         matrix[iMin][j]=matrix[iMax][j];
41.         matrix[iMax][j]=temp;
42.     }
43.
44.
45.     std::cout<<"Отредактированная матрица: " << std::endl;
46.     for(int i=0;i<m;i++){
47.         for(int j=0;j<n;j++){
48.             std::cout << matrix[i][j] << " ";

```

```
49.         }
50.         std::cout << std::endl;
51.     }
52.
53.
54.     for (int i = 0; i < m; i++)
55.         delete[] matrix[i];
56.     delete[] matrix;
57.
58.     return 0;
59. }
```