

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
Инженерно-физический факультет  
Кафедра автоматизированных систем обработки информации и  
управления

ОТЧЕТ ПО ПРАКТИКЕ

Программная реализация ранга матрицы.  
*Вариант 6.*

1 курс, группа ИИВТ1-2

Выполнила:

\_\_\_\_\_ А. Н. Лисова  
«\_\_\_» \_\_\_\_\_ 2023 г.

Руководитель:

\_\_\_\_\_ С. В. Теплоухов  
«\_\_\_» \_\_\_\_\_ 2023 г.

Майкоп, 2023 г.

# 1. Введение

## 1.1. Формулировка цели

Целью данной работы является написание программы для нахождения ранга матрицы.

### 1.1.1. Теория

Нахождение ранга матрицы способом элементарных преобразований (методом Гаусса). Под элементарными преобразованиями матрицы понимаются следующие операции:

- 1) умножение на число, отличное от нуля;
- 2) прибавление к элементам какой-либо строки или какого-либо столбца;
- 3) перемена местами двух строк или столбцов матрицы;
- 4) удаление "нулевых" строк, то есть таких, все элементы которых равны нулю;
- 5) удаление всех пропорциональных строк, кроме одной.

Для любой матрицы  $A$  всегда можно прийти к такой матрице  $B$ , вычисление ранга которой не представляет затруднений. Для этого следует добиться, чтобы матрица  $B$  была трапецевидной. Тогда ранг полученной матрицы будет равен числу строк в ней кроме строк, полностью состоящих из нулей.

Ступенчатую матрицу называют трапецевидной или трапецеидальной, если для ведущих элементов  $a_{1k_1}, a_{2k_2}, \dots, a_{rk_r}$  выполнены условия  $k_1=1, k_2=2, \dots, k_r=r$ , т.е. ведущими являются диагональные элементы. В общем виде трапецевидную матрицу можно записать так:

$$A_{m \times n} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1r} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2r} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{rr} & \dots & a_{rn} \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

## 2. Ход работы

### 2.1. Код выполненной программы

```
#include <iostream>
#include <vector>
```

```

#include <stdlib.h>
#include <Windows.h>
#include <cmath>
using namespace std;

int main()
{
SetConsoleCP(1251);
SetConsoleOutputCP(1251);
int row, col, sum = 0, step = 0, sort1, sort2, rank;
int i, j, k, p, e;
double tempmath, eps = 0.00001;
cout << "Введите количество рядов: ";
cin >> row;
cout << "Введите количество столбцов: ";
cin >> col;
if (row <= 0 || col <= 0)
{
cout << "Ошибка. Неверные параметры матрицы." << endl;
return 0;
}
rank = row;
vector <vector <double>> matrix;

// ввод матрицы
for (int i = 0; i < row; i++)
{
vector <double> temp;
for (int j = 0; j < col; j++)
{
cout << "Введите значение элемента matrix["
<< i << "]"[" << j << "]: ";
cin >> e;
temp.push_back(e);
}
matrix.push_back(temp);
}

// вывод матрицы
cout << endl;
cout << "\nВот как Вы заполнили Вашу матрицу:\n";
for (int i = 0; i < row; i++)
{
for (int j = 0; j < col; j++)

```

```

cout << "[" << i << "]" << "[" << j
<< "]" == " << matrix[i][j] << "\t";
cout << endl;
}
cout << "\n";

// приведение матрицы по методу Гаусса
// проверка условия для диагонали ступеней
if (col > row - 1)
{
for (k = 0; k < row - 1; k++)
{
// перенос нулей столбца в низ матрицы путём перемены строк (сортировка методом пузырька)
j = k;
for (sort1 = k; sort1 < row; sort1++) {
for (sort2 = k; sort2 < row - 1; sort2++) {
if (abs(matrix[sort2][j]) < abs(matrix[sort2 + 1][j]))
{
for (j = 0; j < col; j++) swap(matrix[sort2][j], matrix[sort2 + 1][j]);
}
j = k;
}
}

// преобразования к ступенчатому виду
for (i = k + 1; i < row; i++)
{
j = k;
tempmath = matrix[i][j] / matrix[i - 1 - step][j];
if (matrix[i][j] == 0)
{
step++;
continue;
}
else
{
for (j = k; j < col; j++)
{
matrix[i - step - 1][j] = matrix[i - step - 1][j] * tempmath;
matrix[i][j] = matrix[i][j] - matrix[i - step - 1][j];
}
step++;
}
}
}

```

```

step = 0;

// перенос нулевых строк в самый низ и подсчёт ранга матрицы
for (p = 0; p < col; p++)
if (matrix[k + 1][p] == 0) sum++;
if (sum == col)
{
for (p = 0; p < col; p++)
{
swap(matrix[row - 1][p], matrix[k + 1][p]);
}
}
sum = 0;
}
else
{
for (k = 0; k < col; k++)
{
// перенос нулей столбца в низ матрицы путём перемены строк (сортировка методом пузырька)
j = k;
for (sort1 = k; sort1 < row; sort1++) {
for (sort2 = k; sort2 < row - 1; sort2++) {
if (matrix[sort2][j] < matrix[sort2 + 1][j])
{
for (j = 0; j < col; j++) swap(matrix[sort2][j], matrix[sort2 + 1][j]);
}
}
j = k;
}
}

// преобразования к ступенчатому виду
for (i = k + 1; i < row; i++)
{
j = k;
tempmath = matrix[i][j] / matrix[i - 1 - step][j];
if (matrix[i][j] == 0)
{
step++;
continue;
}
else
{
for (j = k; j < col; j++)

```

```

{
matrix[i - step - 1][j] = matrix[i - step - 1][j] * tempmath;
matrix[i][j] = matrix[i][j] - matrix[i - step - 1][j];
}
step++;
}
}
step = 0;

// перенос нулевых строк в самый низ матрицы
for (p = 0; p < col; p++)
if (matrix[k + 1][p] == 0) sum++;
if (sum == col)
{
for (p = 0; p < col; p++)
{
swap(matrix[row - 1][p], matrix[k + 1][p]);
}
}
sum = 0;
}
}

// округление чисел близких к нулю
for (i = 0; i < row; i++)
{
for (j = 0; j < col; j++)
{
if (abs(matrix[i][j]) < eps) matrix[i][j] = 0;
}
}

// вывод приведенной матрицы
cout << "\nПриведенная матрица:" << endl;
for (int i = 0; i < row; i++)
{
for (int j = 0; j < col; j++)
cout << "[" << i << "]" << "[" << j
<< "]" == " << matrix[i][j] << "\t";
cout << endl;
}
cout << "\n";

// получение ранга матрицы

```

```

for (i = 0; i < row; i++)
{
for (j = 0; j < col; j++)
{
if (matrix[i][j] == 0) sum++;
}
if (sum == col) rank--;
sum = 0;
}

// вывод ранга матрицы
cout << "\nРанг матрицы: " << rank << endl;
return 0;
}

```

```

Введите количество рядов: 3
Введите количество столбцов: 4
Введите значение элемента matrix[0][0]: 1
Введите значение элемента matrix[0][1]: 3
Введите значение элемента matrix[0][2]: 6
Введите значение элемента matrix[0][3]: 7
Введите значение элемента matrix[1][0]: 8
Введите значение элемента matrix[1][1]: 9
Введите значение элемента matrix[1][2]: 3
Введите значение элемента matrix[1][3]: 6
Введите значение элемента matrix[2][0]: 8
Введите значение элемента matrix[2][1]: 3
Введите значение элемента matrix[2][2]: 8
Введите значение элемента matrix[2][3]: 4

Вот как Вы заполнили Вашу матрицу:
[0][0] == 1    [0][1] == 3    [0][2] == 6    [0][3] == 7
[1][0] == 8    [1][1] == 9    [1][2] == 3    [1][3] == 6
[2][0] == 8    [2][1] == 3    [2][2] == 8    [2][3] == 4

Приведенная матрица:
[0][0] == 1    [0][1] == 1.125 [0][2] == 0.375 [0][3] == 0.75
[1][0] == 0    [1][1] == 1.875 [1][2] == -1.5625 [1][3] == 0.625
[2][0] == 0    [2][1] == 0    [2][2] == 7.1875 [2][3] == 5.625

Ранг матрицы: 3

C:\Users\GE70\Desktop\университет\программирование\практика\Practice\Debug\Practice.exe (процесс 6796) завершает работу
с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" ->
"Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рис. 1. Результат работы