

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО
ПРИБОРОСТРОЕНИЯ»
(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент

Л.Н. Бариков

Отчет

по лабораторной работе № 9

по дисциплине Информатика

на тему: «Статические двумерные массивы»

Работу выполнил

студент гр. 4342

С.М.Иванов

Санкт-Петербург

2024

16.

В заданной квадратной матрице размера $(2n+1) \times (2n+1)$ поменять местами среднюю строку и средний столбец.

код на гитхабе для более удобного просмотра:

https://github.com/crimson-catfish/SUAI_labs/blob/main/code/lab9.cpp

программа считывает квадратную матрицу у пользователя, меняет местами среднюю строку и столбец матрицы, а затем печатает исходную и замененную матрицы. Программа продолжает работать до тех пор, пока пользователь не решит выйти.

разбивка кода:

Программа начинается с включения библиотеки `iostream` и определения константы `MATRIX_SIDE` как 16. Однако эта константа не используется в коде.

Тип `my_element` определяется как псевдоним типа `int`, а тип матрицы определяется как двумерный массив `my_element` со строками и столбцами `MATRIX_SIDE`.

Функция `read_width()` считывает половину ширины матрицы от пользователя и возвращает удвоенное введенное значение плюс один. Это гарантирует, что ширина матрицы всегда будет нечетной.

Функция `read_matrix()` считывает элементы матрицы от пользователя и сохраняет их в матрице `m`.

Функция `swap_middle_row_and_column()` меняет местами среднюю строку и столбец матрицы.

Функция `print_matrix()` печатает элементы матрицы в отформатированном виде.

Функция main() является точкой входа в программу. Он неоднократно считывает ширину и элементы матрицы от пользователя, меняет местами среднюю строку и столбец и печатает исходную и замененную матрицы. Программа продолжает работать до тех пор, пока пользователь не введет букву, отличную от «R» или «r».

```
Enter half of matrix width: 1
```

```
Enter matrix 3x3
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
Original matrix:
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
Swaped matrix:
```

```
1 4 3
```

```
2 5 8
```

```
7 6 9
```

```
Enter "R" to run again, or enter any other letter to quit programm: q
```

```
Enter half of matrix width: 2
```

```
Enter matrix 5x5
```

```
1 -2 5 7 8 -1 8 32 32 -12 3 78 0 7 -2 -53 78 9 -3 1 78 9 3 -34 69
```

```
Original matrix:
```

```
1 -2 5 7 8
```

```
-1 8 32 32-12
```

```
3 78 0 7 -2
```

```
-53 78 9 -3 1
```

```
78 9 3-34 69
```

```
Swaped matrix:
```

```
1 -2 3 7 8
```

```
-1 8 78 32-12
```

```
5 32 0 9 3
```

```
-53 78 7 -3 1
```

```
78 9 -2-34 69
```

```
Enter "R" to run again, or enter any other letter to quit programm: q
```

```
#include <iostream>
```

```
using namespace std;
```

```
const int MATRIX_SIDE = 16;
```

```
typedef int my_element;
```

```
typedef my_element matrix[MATRIX_SIDE][MATRIX_SIDE];
```

```
typedef unsigned int matrix_width;
```

```
matrix_width read_width()
```

```
{
```

```
cout << "Enter half of matrix width: ";
```

```
unsigned int n;
```

```
cin >> n;
```

```
return 2 * n + 1;
```

```
}
```

```
void read_matrix(matrix_width width, matrix m)
```

```
{
```

```
cout << "Enter matrix " << width << 'x' << width << "\n";
```

```
for (int y = 0; y < width; y++)
```

```
{
```

```
for (int x = 0; x < width; x++)
```

```
cin >> m[y][x];
```

```
}
```

```
}
```

```
void swap_middle_row_and_column(matrix_width width, matrix m)
```

```
{
```

```
my_element buffer;
```

```
for (int i = 0; i < width; i++)
```

```
{
```

```
buffer = m[width / 2][i];
```

```
m[width / 2][i] = m[i][width / 2];
```

```
m[i][width / 2] = buffer;
```

```
}
```

```
}
```

```
void print_matrix(matrix_width width, matrix m)
```

```
{
```

```
for (int i = 0; i < width; i++)
```

```
{
```

```
cout << endl;
for (int j = 0; j < width; j++)
{
    cout.width(3);
    cout << m[i][j];
}
}
}
```

```
int main()
```

```
{
    char quitOrRestart = 'r';
```

```
do
```

```
{
    matrix_width width = read_width();
    matrix m;
    read_matrix(width, m);
```

```
    cout << "\n\nOriginal matrix:";
    print_matrix(width, m);
```

```
    swap_middle_row_and_column(width, m);
```

```
    cout << "\n\nSwaped matrix:";
    print_matrix(width, m);
```

```
    cout << "\n\nEnter \"R\" to run again, or enter any other letter to quit programm: ";
    cin >> quitOrRestart;
} while (quitOrRestart == 'r' || quitOrRestart == 'R');
```

```
return 0;
}
```