

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

Функция `read_width()` вызывается для считывания половины ширины матрицы от пользователя. Затем он возвращает удвоенное введенное значение плюс один, чтобы гарантировать нечетность ширины матрицы.

Функция `read_matrix(unsigned int width)` вызывается для чтения элементов матрицы из пользовательского ввода. Он создает двумерный массив целых чисел заданной ширины, считывает элементы построчно и возвращает матрицу.

Функция `swap_middle_row_and_column(int ** m, unsigned int width)` вызывается для замены средней строки и столбца матрицы. Он перебирает матрицу, заменяя элементы в средней строке соответствующими элементами в среднем столбце.

Функция `print_matrix(int ** m, unsigned int width)` вызывается для печати элементов матрицы. Он перебирает матрицу и печатает каждый элемент с интервалом табуляции.

В функции `main()` цикл `do-while` используется для многократного чтения матриц, замены их средней строки и столбца и печати результатов до тех пор, пока пользователь не решит выйти из программы. Пользователю предлагается ввести «R» для повторного запуска или любую другую букву для выхода из программы.

Алгоритм использует указатели и динамическое распределение памяти для создания матрицы и управления ею. Функция `read_matrix()` выделяет память для матрицы, а функция `swap_middle_row_and_column()` изменяет матрицу на месте.

Функция `print_matrix()` используется для отображения содержимого матрицы на экране.

```
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: r
```

Enter half of matrix width: 2

Enter matrix 5x5

1 -2 5 7 8 -1 8 32 32 -12 -5 0 0 23 8 1 2 3 4 5 7 6 5 90 -2

Original matrix:

1	-2	5	7	8
-1	8	32	32	-12
-5	0	0	23	8
1	2	3	4	5
7	6	5	90	-2

Swaped matrix:

1	-2	-5	7	8
-1	8	0	32	-12
5	32	0	3	5
1	2	23	4	5
7	6	8	90	-2

```
#include <iostream>
```

```
using namespace std;
```

```
unsigned int read_width()
```

```
{
```

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

```
unsigned int n;
```

```
cin >> n;
```

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

```
}
```

```

int ** read_matrix(unsigned int width)
{
    int ** m;
    m = new int * [width];
    cout << "Enter matrix " << width << 'x' << width << "\n";
    for (int y = 0; y < width; y++)
    {
        m[y] = new int [width];
        for (int x = 0; x < width; x++)
        {
            cin >> m[y][x];
        }
    }
    return m;
}

```

```

int ** swap_middle_row_and_column(int ** m, unsigned int width)
{
    unsigned int 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;
    }
    return m;
}

```

```

void print_matrix(int ** m, unsigned int width)
{
    for (int i = 0; i < width; i++)
    {
        cout << endl << endl;
        for (int j = 0; j < width; j++)
        {
            cout << m[i][j] << '\t';
        }
    }
}

```

```

int main()
{

```

```
char quitOrRestart = 'r';

do
{
    unsigned int width = read_width();
    int ** m = read_matrix(width);

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

    m = swap_middle_row_and_column(m, width);

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

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

return 0;
}
```