

Лекция 16

***Основы программирования на C++.
Двумерные массивы. Продолжение***

Создать и вывести массив целых чисел из диапазона [-5;10]. Вставить в массив 5 столбец после третьего столбца.

1 2 3 4 5 6

4 5 6 1 1 2

7 8 9 0 1 2

1 2 3 5 4 5 6

4 5 6 1 1 1 2

7 8 9 1 0 1 2

При объявлении массив задать на 1 столбец больше!!!

A[n][m+1];

Освободить место при помощи сдвига ВПРАВО, цикл от КОНЦА массива

X=A[3]; - одномерном

int x[n] – массив-столбец

```
<iostream>
<cstdlib>
<time.h>
<iomanip>
```

```
using namespace std;
```

```
// ВСТАВИТЬ И ВЫВЕСТИ МАССИВ ЦЕЛЫХ ЧИСЕЛ ИЗ ДИАПАЗОНА [-5;
```

```
int argc, char** argv)
```

```
locale(0, "");
```

```
time_t t = time(0);
```

```
int n, m, i, j;
```

```
cout << "СКОЛЬКО СТРОК?" << endl;
```

```
>> n;
```

```
cout << "СКОЛЬКО СТОЛБЦОВ?" << endl;
```

```
>> m;
```

```
int A[n][m+1];
```

```
cout << "ИСХОДНЫЙ МАССИВ" << endl;
```

```
for (i=0; i<n; i++)
```

```
for (j=0; j<m; j++)
```

```
{
```

```
    A[i][j] = rand() % 16 - 5;
```

```
    cout << setw(4) << A[i][j];
```

```
}
```

```
cout << endl;
```

```
// ВСТАВИТЬ В МАССИВ 5 СТОЛБЕЦ ПОСЛЕ ТРЕТЬЕГО СТОЛБЦА.
```

```
int x[n];
```

```
// ЗАПОМНИТЬ НУЖНЫЙ СТОЛБЕЦ (ТОТ КОТОРЫЙ ТРЕБУЕТСЯ ВСТАВИТЬ)
```

```
for (i=0; i<n; i++)
```

```
    x[i] = A[i][4];
```

```
// СДВИГ СТОЛБЦОВ
```

```
int nom = 2;
```

```
for (i=0; i<n; i++)
```

```
    for (j = m; j > nom; j--)
```

```
        A[i][j] = A[i][j-1];
```

```
// ВСТАВКА
```

```
for (i=0; i<n; i++)
```

```
    A[i][nom+1] = x[i];
```

```
cout << "ПОЛУЧЕННЫЙ МАССИВ В КОТОРОМ ВСТАВИЛИ 5 СТОЛБЕЦ ПОСЛЕ ТРЕТЬЕГО" << endl;
```

```
for (i=0; i<n; i++)
```

```
{
```

```
    for (j=0; j<m+1; j++)
```

```
    {
```

```
        cout << setw(4) << A[i][j];
```

```
    }
```

```
    cout << endl;
```

```
}
```

Вставить строку с минимальным четным элементом после первой строки.

```
<iostream>
<cstdlib>
<time.h>
<iomanip>

namespace std;

// Вставить строку с минимальным четным элементом после первой
// строки и вывести массив целых чисел
int main(int argc, char** argv) {
    srand(time(0));
    int n, m, i, j;
    cout << "сколько строк?" << endl;
    n = 10;
    cout << "сколько столбцов?" << endl;
    m = 10;
    int A[n+1][m];
    cout << "исходный массив" << endl;
    for(i=0; i<n; i++)
        for(j=0; j<m; j++)
            A[i][j] = rand() % 16;
    cout << endl;

    // Вставить строку с минимальным четным элементом после первой
    // строки
    int min, nommin;
    for(i=0; i<n; i++)
        for(j=0; j<m; j++)
            if (A[i][j] % 2 == 0)
                if (A[i][j] < min) {
                    min = A[i][j];
                    nommin = i;
                }

    // Вставить строку с минимальным четным элементом после первой
    // строки
    int x[m]; // массив-строка
    // запомнить нужную строку (ту которую требуется вставить)
    for (j=0; j<m; j++)
        x[j] = A[nommin][j];

    // сдвиг строк
    int nom = 0; // номер строки после которой вставить строку
    for(i = n; i > nom; i--)
        for (j = 0; j < m; j++)
            A[i][j] = A[i-1][j];

    // вставка
    for (j=0; j<m; j++)
        A[nom+1][j] = x[j];

    cout << "полученный массив в котором вставили строку" << endl;
    for(i=0; i<n+1; i++)
    {
        for(j=0; j<m; j++)
        {
            cout << setw(4) << A[i][j];
        }
        cout << endl;
    }

    return 0;
}
```

Поменять в массиве местами столбцы с минимальным и максимальным элементами.

//Поменять в массиве местами столбцы с минимальным и максимальным элементами

```
int min=A[0][0],max=A[0][0],nommax=0,nommin=0;
```

```
for(i=0; i<n; i++)
```

```
    for(j=0; j<m; j++)
```

```
    {
```

```
        if(A[i][j]<min)
```

```
        {
```

```
            min=A[i][j];
```

```
            nommin=j;
```

```
        }
```

```
        if(A[i][j]>max)
```

```
        {
```

```
            max=A[i][j];
```

```
            nommax=j;
```

```
        }
```

```
    }
```

```
cout<<"nommin = "<<nommin+1<<endl;
```

```
cout<<"nommax = "<<nommax+1<<endl;
```

```
for(i=0; i<n; i++)
```

```
{
```

```
    int x=A[i][nommax];
```

```
    A[i][nommax] = A[i][nommin];
```

```
    A[i][nommin] = x;
```

```
}
```

Квадратные массивы

В математике **квадратная матрица** — это матрица, у которой число строк совпадает с числом столбцов, и это число называется порядком **матрицы**.

int A[n][n];

Главной диагональю матрицы называется диагональ идущая из левого верхнего угла в правый нижний угол.

$$i = j$$

A[0][0], A[1][1], A[2][2], ...

Побочной диагональю матрицы называется диагональ идущая из левого нижнего угла в правый верхний угол.

$$i + j = n - 1$$

A[0][n-1], A[1][n-2], A[2][n-3], ...

2	3	5
-1	4	0
3	-2	1

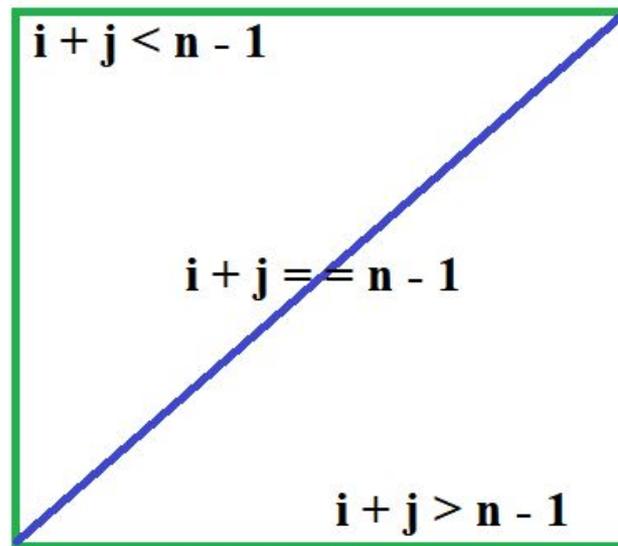
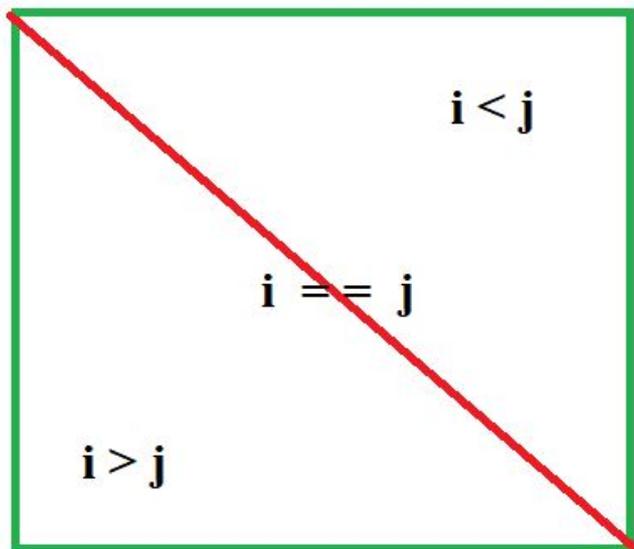
2	3	5
-1	4	0
3	-2	1

главная

выше главной

выше побочной

побочная



ниже главной

ниже побочной

2	3	5
-1	4	0
3	-2	1

2	3	5
-1	4	0
3	-2	1

Вывести элементы на главной диагонали, на побочной диагонали.
Найти сумму элементов ниже главной диагонали. Найти произведение элементов выше побочной диагонали.

```
#include <iostream>
#include <cstdlib>
#include <time.h>
#include <iomanip>

using namespace std;

//Создать и вывести массив целых чисел из д
int main(int argc, char** argv)
{
    setlocale(0, "");
    srand(time(0));
    int n, i, j;
    cout<<"порядок квадратной матрицы?"<<en
    cin >> n;
    int A[n][n];
    cout<<"исходный массив"<<endl;
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            A[i][j]=rand()%16 - 5;
            cout<<setw(4)<<A[i][j];
        }
        cout<<endl;
    }
}
```

```
// Вывести элементы на главной диагонали, на побочной д
cout<<"элементы главной диагонали:"<<endl;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i==j) cout<<setw(4)<<A[i][j];
cout<<endl;

cout<<"элементы побочной диагонали:"<<endl;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i+j==n-1) cout<<setw(4)<<A[i][j];
cout<<endl;

//Найти сумму элементов ниже главной диагонали.
int s=0;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i>j) s+=A[i][j];
cout<<"сумму элементов ниже главной диагонали = "<<s;

//Найти произведение элементов выше побочной диагонали.
int p=1;
for(i=0; i<n; i++)
    for(j=0; j<n; j++)
        if (i+j<n-1) p*=A[i][j];
cout<<"произведение элементов выше побочной диагонали"
```

Заполнение массива по формуле $i+j$

```
for (i=1; i<n; i++)  
    for (j=1; j<n; j++)  
        A[i][j]=i+j;
```

```
for (i=1; i<n; i++)  
{  
    for (j=1; j<n; j++)  
    {  
        cout<<setw(4)<<A[i][j];  
    }  
    cout<<endl;  
}
```