

Программирование на языке Си#

Тема 8. Массивы (двумерные)

Матрицы

Матрица – это прямоугольная таблица чисел (или других элементов одного типа).

Матрица – это массив, в котором каждый элемент имеет два индекса (номер строки и номер столбца).

A

	1	2	3	4	5
1	1	4	7	3	6
2	2	-5	0	15	10
3	8	9	11	12	20

столбец 3

строка 2

ячейка **A**[3, 4]

Объявление массивов

Массив целых чисел:

ИМЯ

Количество
строк 4
Столбцов 5

```
int[, ] A = new int [4,5];
```

Значение всех
Элементов
(по умолчанию!)

0

ТИП
элементов

Объявление массивов

Массив целых чисел (квадратная матрица $n \times n$):

ИМЯ

Количество
строк n
Столбцов n

```
int[, ] A = new int [n,n];
```

Значение всех
Элементов
(по умолчанию!)

0

ТИП
элементов

Объявление массивов

Массив целых чисел (матрица с размером $n*m$):

ИМЯ

Количество строк n
Столбцов m

```
int[, ] A = new int [n,m];
```

Значение всех
Элементов
(по умолчанию!)

0

ТИП
элементов

Три этапа работы с массивами

1. Заполнение массива данными

3. Вывод массива или отдельных данных на экран

Способы ввода данных в массив:

- 1. С клавиатуры (2 способа!);***
- 2. С помощью генератора случайных чисел;***
- 3. С помощью вычисления элементов.***

1. С клавиатуры в столбик

```
for (i = 0; i < n; i++)  
{  
    for (j = 0; i < n; j++)  
    {  
        m[i,j] = Convert.ToInt32(Console.ReadLine());  
    }  
}
```

Удобно?

2. С клавиатуры квадратной матрицей

```
int i, n, j;
n = Convert.ToInt32(Console.ReadLine());
int[,] m = new int[n, n];
string s;
for (i = 0; i < n; i++)
{
    s = Console.ReadLine();
    string[] arr = s.Split();

    for (j = 0; j < n; j++)
    {
        m[i, j] = Convert.ToInt32(arr[j]);
    }
}
```

С помощью генератора случайных чисел

```
Random rand = new Random();  
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < n; j++)  
    {  
        m[i,j] = rand.Next(-100, 100);  
    }  
}
```

Способ вывода данных из массива:

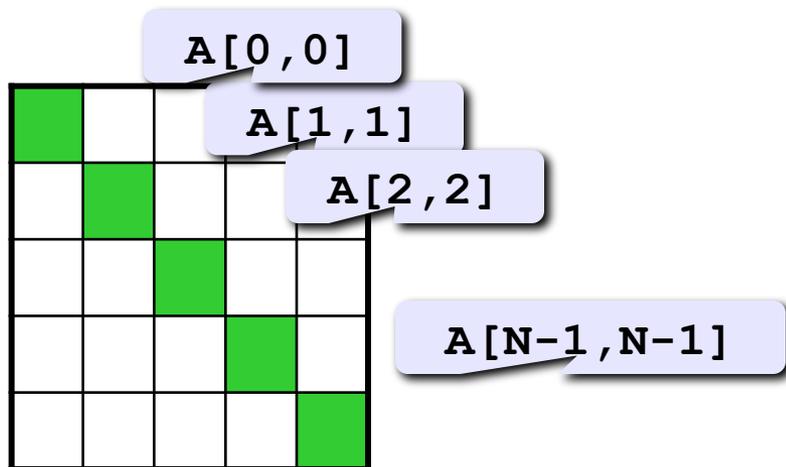
```
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < n; j++)  
    {  
        Console.Write(m[i,j]+“ “);  
    }  
    Console.WriteLine();  
}
```

**Вывод одной
строки**

**Переход на
новую строку**

Операции с матрицами

Задача 1. Вывести на экран главную диагональ квадратной матрицы из N строк и N столбцов.

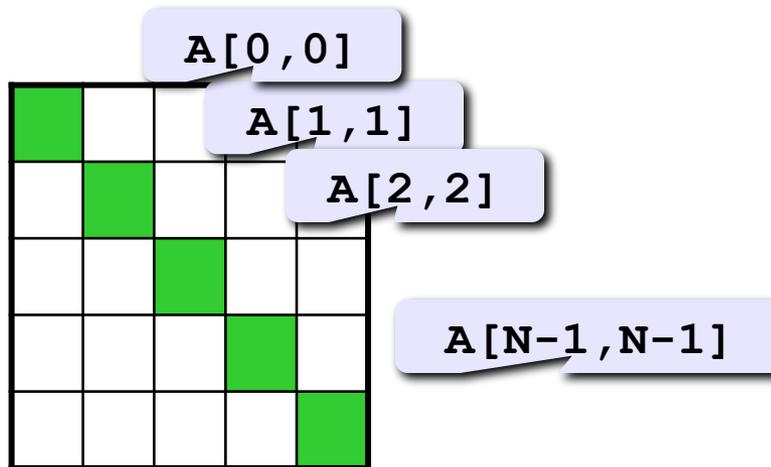


Какое условие?

Номер строки равен номеру столбца

Операции с матрицами

Задача 1. Вывести на экран главную диагональ квадратной матрицы из N строк и N столбцов.



1 способ:

```
for (i = 0; i < n; i++)  
for (j = 0; j < n; j++)  
{  
    if ( i==j )  
    {  
        Console.Write(m[i,j]);  
    }  
}
```

2 способ:

```
for (i = 0; i < n; i++)  
    Console.Write(m[i,i]);
```

Операции с матрицами

Задача 1. Вывести на экран главную диагональ квадратной матрицы из N строк и N столбцов.

1. Ввод n

2. Заполнить матрицу

3. Вывести диагональ

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

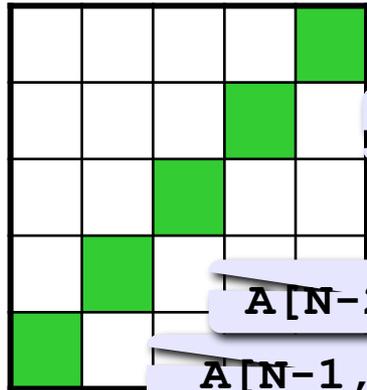
```
for (j = 0; j < n; j++)
```

```
{  
    if ( i==j )  
    {  
        Console.Write(m[i,j]);  
    }  
}
```

```
int i, n, j;  
n = Convert.ToInt32(Console.ReadLine());  
int[,] m = new int[n, n];  
string s;  
for (i = 0; i < n; i++)  
{  
    s = Console.ReadLine();  
    string[] arr = s.Split();  
  
    for (j = 0; j < n; j++)  
    {  
        m[i, j] = Convert.ToInt32(arr[j]);  
    }  
}
```

Операции с матрицами

Задача 2. Вывести на экран вторую диагональ.



$A[0, N-1]$

$A[1, N-2]$

$A[N-2, 1]$

$A[N-1, 0]$

сумма номеров строки и столбца $N-1$?

1 способ:

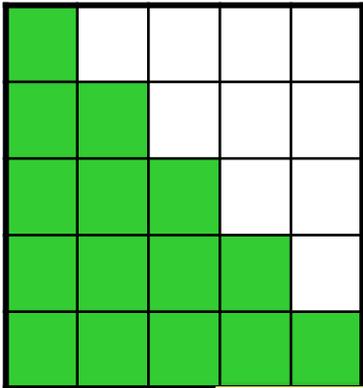
```
for (i = 0; i < n; i++)
for (j = 0; j < n; j++)
{
    if ( i+j==n-1 )
    {
        Console.Write(m[i,j]);
    }
}
```

2 способ:

```
for (i = 0; i < n; i++)
    Console.Write(m[i,n-1-i]);
```

Операции с матрицами

Задача 3. Найти сумму элементов, стоящих на главной диагонали и ниже ее.



строка 0: $A[0, 0]$

строка 1: $A[1, 0] + A[1, 1]$

строка 2: $A[2, 0] + A[2, 1] + A[2, 2]$

...

строка N-1: $A[N-1, 0] + A[N-1, 1] + \dots + A[N-1, N-1]$

1 способ:

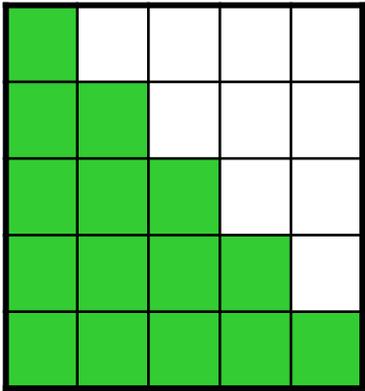
```
S = 0;  
for (i = 0; i < n; i++)  
for (j = 0; j < n; j++)  
{ if (j <= i)  
    S = S + A[i, j];  
}
```

циклы по всем элементам

складываем
нужные элементы

Операции с матрицами

Задача 3. Найти сумму элементов, стоящих на главной диагонали и ниже ее.



строка 0: $A[0, 0]$

строка 1: $A[1, 0] + A[1, 1]$

строка 2: $A[2, 0] + A[2, 1] + A[2, 2]$

...

строка N-1: $A[N-1, 0] + A[N-1, 1] + \dots + A[N-1, N-1]$

2 способ:

```
S = 0;
```

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

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

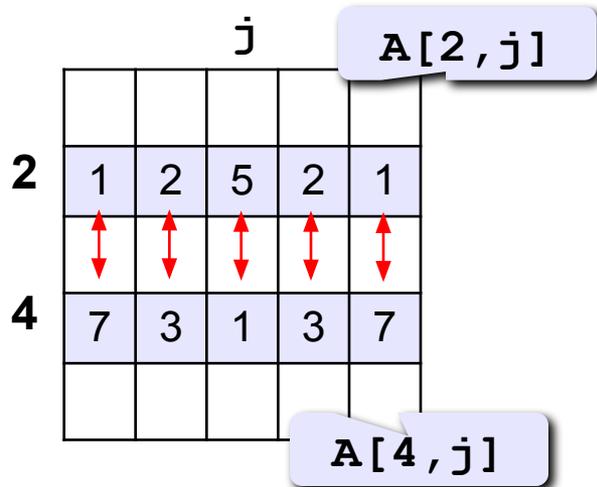
```
        S = S + A[i, j];
```

цикл по всем
строкам

складываем
нужные элементы
строки i

Операции с матрицами

Задача 4. Перестановка строк или столбцов. В матрице из N строк и M столбцов переставить 2-ую и 4-ую строки.



Цикл

```
{  
    c := A[2, j];  
    A[2, j] := A[4, j];  
    A[4, j] := c;  
}
```

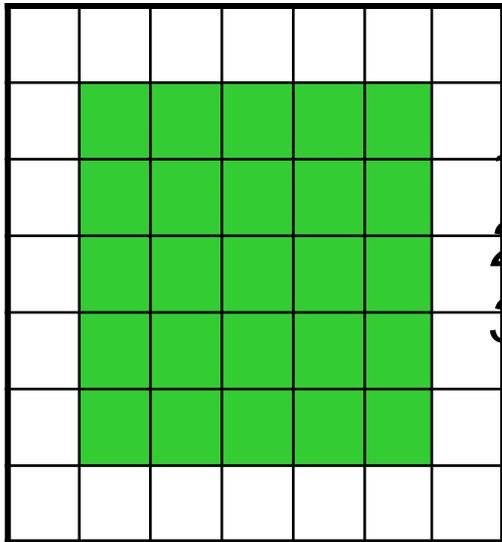
Задача 5. К третьему столбцу добавить шестой.

ЦИКЛ

```
A[i, 3] := A[i, 3] + A[i, 6];
```


Задания

Заполнить матрицу из 7 строк и 7 столбцов случайными числами в интервале $[10,90]$ и вывести ее на экран. Заполнить элементы, отмеченные зеленым фоном, числами 99, и вывести полученную матрицу на экран.



Алгоритм

1. Заполнить массив случайными числами
2. Вывести весь массив
3. Заполнить массив
от $i=1$ до $n-1$ строки
от $j=1$ до $n-1$ столбца
 $m[i,j] = 99$
4. Вывести весь массив

1. Заполнить массив m:

```
Int n=7;  
int[,] m = new int [n,n];  
Random rand = new Random();  
For (i=0; i<n; i++)  
{  
For (j=0; j<n; j++)  
    m[i,j] = rand.Next(10, 90);  
}
```

2. Вывести массив m:

```
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < n; j++)  
    {  
        Console.Write(m[i,j]+" ");  
    }  
    Console.WriteLine();  
}
```

3. Заполнить новую часть массива m:

```
for (i = 1; i < n-1; i++)  
{  
    for (j = 1; j < n-1; j++)  
        {  
            m[i,j]=99  
        }  
}
```

4. Вывести новый массив m:

```
for (i = 0; i < n; i++)  
{  
    for (j = 0; j < n; j++)  
    {  
        Console.Write(m[i,j]+“ “);  
    }  
    Console.WriteLine();  
}
```