

# Двумерные массивы

Обработка  
относительно  
диагоналей.

Практическая работа

1. Составить любые 2 программы по

# Обработка относительно диагоналей

Рассмотрим работу с «квадратными» массивами.

*Это такие массивы, в которых количество столбцов равно количеству строк.*

	1	2	3	4	5	6	7
1	97	5	40	89	3	74	16
2	49	74	45	63	99	61	96
3	37	67	42	65	5	59	7
4	89	49	45	32	57	77	34
5	62	13	93	83	26	10	83
6	83	69	88	30	33	78	21
7	33	12	4	66	98	50	40

	1	2	3	4	5	6	7
1	71	56	8	49	27	70	57
2	32	27	34	30	19	76	81
3	93	81	4	46	70	6	10
4	90	99	7	29	35	49	66
5	22	54	63	37	8	8	59
6	33	74	60	25	18	45	98
7	3	13	43	75	88	38	17

Элементы главной диагонали можно записать так:

$A[i,i]$ ,

т.е. индексы строки и столбца совпадают

Элементы побочная диагонали можно записать так:

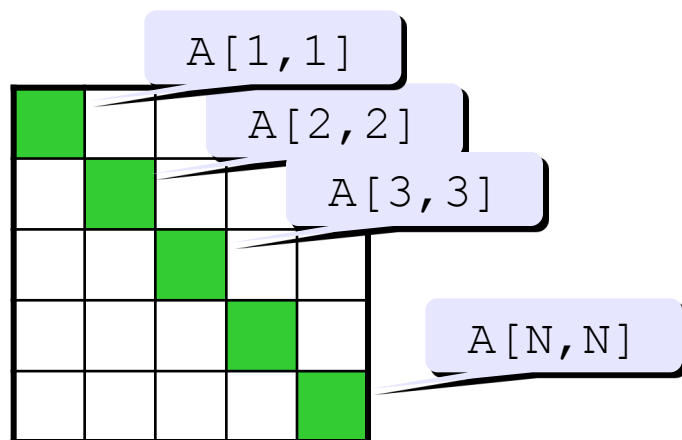
$A[i,n-i+1]$ ,

где  $n$  – размер массива.

	1	2	3	4	5	6	7
1	71	56	8	49	27	70	57
2	32	27	34	30	19	76	81
3	93	81	4	46	70	6	10
4	90	99	7	29	35	49	66
5	22	54	63	37	8	8	59
6	33	74	60	25	18	45	98
7	3	13	43	75	88	38	17

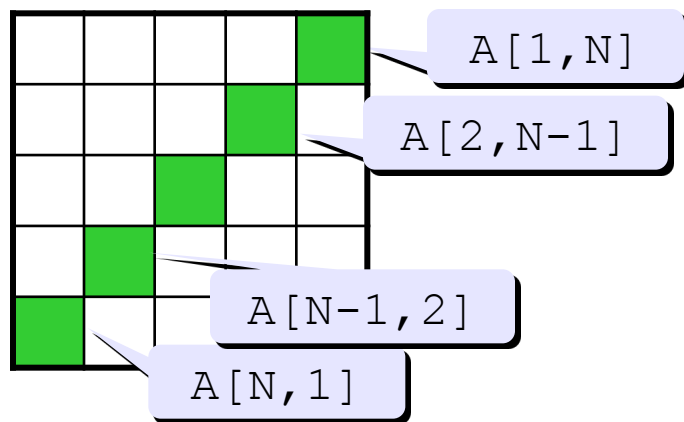
# Операции с матрицами

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



```
for i:=1 to N do
  write ( A[i,i]:5 );
```

**Задача 2.** Вывести на экран побочную диагональ.



сумма номеров строки и столбца  $N+1$

```
for i:=1 to N do
  write ( A[i, N+1-i]:5 );
```

## Обработка относительно диагоналей

Заполнение нулями главной диагонали

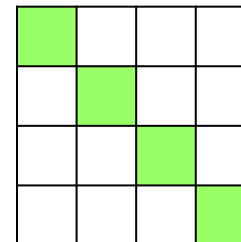
```
for i:=1 to n do  
    a[i,i]:=0;
```

	1	2	3	4	5	6	7
1	<b>0</b>	56	8	49	27	70	57
2	32	<b>0</b>	34	30	19	76	81
3	93	81	<b>0</b>	46	70	6	10
4	90	99	7	<b>0</b>	35	49	66
5	22	54	63	37	<b>0</b>	8	59
6	33	74	60	25	18	<b>0</b>	98
7	3	13	43	75	88	38	<b>0</b>

# Перебор элементов матрицы

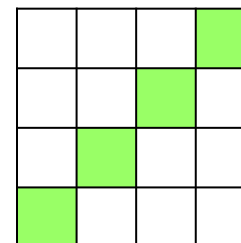
## Главная диагональ:

```
for i:=1 to N do begin
  { работаем с A[i,i] }
end;
```



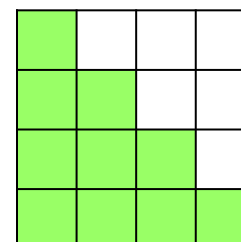
## Побочная диагональ:

```
for i:=1 to N do begin
  { работаем с A[i,N+1-i] }
end;
```



## Главная диагональ и под ней:

```
for i:=1 to N do
  for j:=1 to i do begin
    { работаем с A[i,j] }
  end;
```

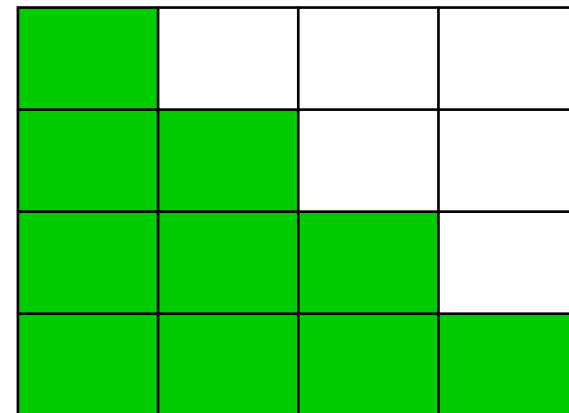


## Обработка относительно диагоналей

---

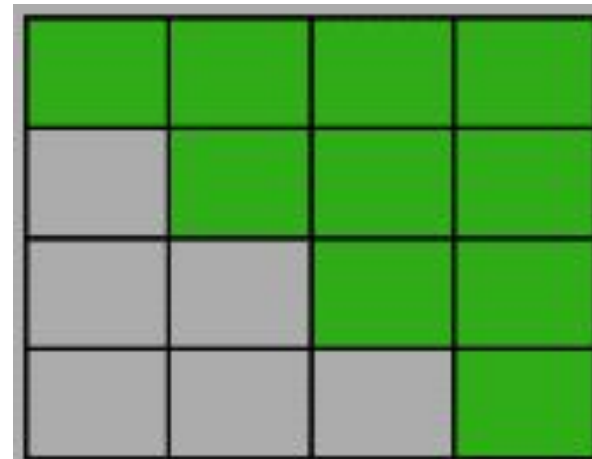
**НИЖЕ** и **НА** главной диагонали:

```
FOR i := 1 TO n DO  
  FOR j := 1 TO i DO
```



**ВЫШЕ** и **НА** главной диагонали:

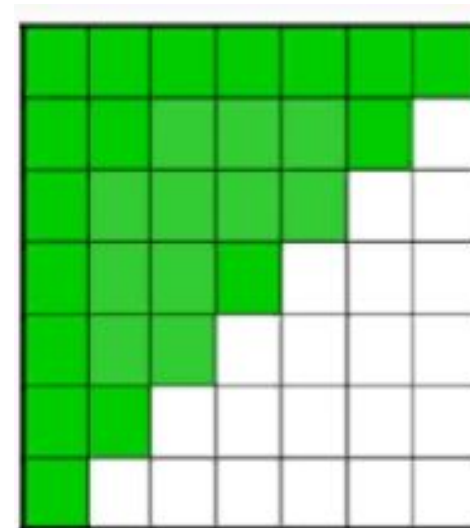
```
FOR i := 1 TO n DO  
  FOR j := i TO n DO
```



## Обработка относительно диагоналей

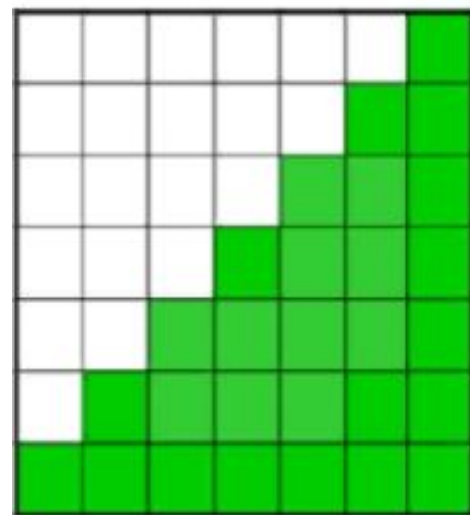
**ВЫШЕ и НА побочной диагонали:**

```
FOR i:=1 TO n DO  
  FOR j:= 1 TO n - i + 1 DO
```



**НИЖЕ и НА побочной диагонали:**

```
FOR i:=1 TO n DO  
  FOR j:= n - i + 1 TO n DO
```





# Обработка относительно диагоналей

## Заполнение нулями элементов выше побочной диагонали

```
for i:=1 to n do  
  for j:=1 to n-i do  
    a[i,j]:=0;
```

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	89
2	0	0	0	0	0	2	34
3	0	0	0	0	78	92	12
4	0	0	0	9	58	39	7
5	0	0	91	34	67	20	87
6	0	81	35	61	22	39	96
7	43	45	94	74	53	47	23

# Обработка относительно диагоналей

Заполнение нулями элементов по периметру,  
массива

```

for i:=1 to n do
  begin
    a[i,1]:=0;
    a[i,n]:=0;
    a[1,i]:=0;
    a[n,i]:=0;
  end;

```

1	0	0	0	0	0	0	0
2	0	76	91	18	68	81	0
3	0	5	40	88	91	35	0
4	0	67	55	9	34	61	0
5	0	93	78	58	67	22	0
6	0	2	92	39	20	39	0
7	0	0	0	0	0	0	0

## Задачи

---

1. Создать квадратную матрицу случайных чисел. Найти сумму элементов ее главной и побочной диагонали.

2. Заполнить элементы квадратного массива так, как предложено на рисунке:

а) 100000000

110000000

111000000

111100000

111110000

111111000

111111100

111111110

111111111

б) 111111111

011111111

001111111

000111111

000011111

000001111

000000111

000000011

000000001

в) 000000001

000000011

000000111

000001111

000011111

000111111

001111111

011111111

111111111

# Задачи

---

г) 111111111

111111110

111111100

111111000

111110000

111100000

111000000

110000000

000000000

д) 100000000

110000000

111000000

111100000

111110000

111100000

111000000

110000000

000000000

е) 000000001

000000011

000000111

000001111

000011111

000001111

000000111

000000011

000000000

# Задачи

---

ж) 11111111

01111110

00111100

000111000

000010000

000000000

000000000

000000000

000000000

з) 00000000

00000000

00000000

00000000

000010000

000111000

001111100

011111110

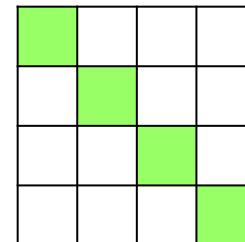
11111111

# Программирование на языке Python

# Перебор элементов матрицы

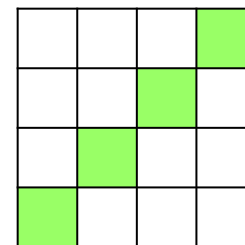
## Главная диагональ:

```
for i in range(N) :  
    # работаем с A[i][i]
```



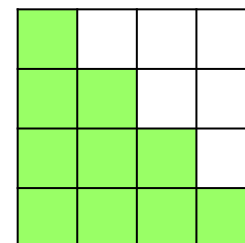
## Побочная диагональ:

```
for i in range(N) :  
    # работаем с A[i][N-1-i]
```



## Главная диагональ и под ней:

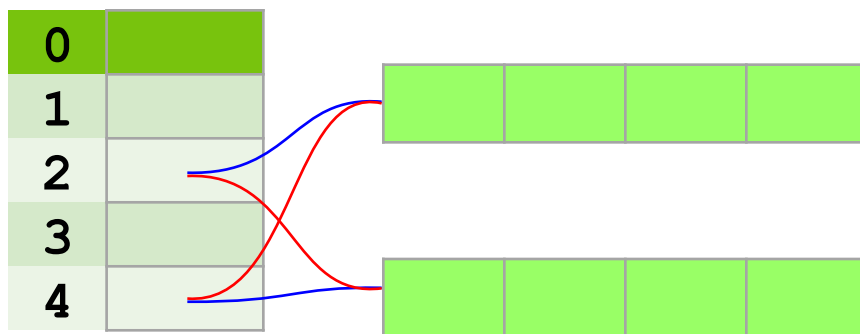
```
for i in range(N) :  
    for j in range(i+1):  
        # работаем с A[i][j]
```



# Перестановка строк и столбцов

2-я и 4-я строки:

```
A[2], A[4] = A[4], A[2]
```



2-й и 4-й столбцы:

```
for i in range(N):  
    A[i][2], A[i][4] = A[i][4], A[i][2]
```



# Выделение строк и столбцов

---

1-я строка:

```
R = A[1][:]
```

```
R = A[i]
```

2-й столбец:

```
C = []  
for row in A:  
    C.append(row[2])
```

или так:

```
C = [ row[2] for row in A ]
```

главная диагональ:

```
D = [ A[i][i] for i in range(N) ]
```