

# Модуль 5. Массивы

---

**Рассматриваются массивы, способы их описания, формирования и обработки**

# Описание массивов. Одномерные массивы.

## Пример 1.

```
float p[10];
int a[5] = {10, 20, 30, 40, 50};
int b[5]={3,2,1};
// b[0]=3, b[1]=2, b[2]=1, b[3]=0, b[4]=0
char cv[4] = { 'a', 's', 'd', 'f' };
```

a[55] – индекс задается как константа,  
a[I] – индекс задается как переменная,  
a[2\*I] – индекс задается как выражение.

```
#include <iostream.h>
```

```
int main() {
```

```
    const int n = 10;
```

```
    int marks[n] = {3, 4, 5, 4, 4};
```

```
    int i, sum;
```

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

```
        sum += marks[i];
```

```
    cout << "Сумма элементов: " << sum;
```

```
}
```

Память



Адрес  
начала  
массива

# Перебор массивов по одному элементу

**Элементы можно перебирать:**

1) Слева направо с шагом 1, используя цикл с параметром

```
for(int I=0;I<n;I++){обработка a[I];}
```

2) Слева направо с шагом отличным от 1, используя цикл с параметром

```
for (int I=0;I<n;I+=step){обработка a[I];}
```

3) Справа налево с шагом 1, используя цикл с параметром

```
for(int I=n-1;I>=0;I--){обработка a[I];}
```

4) Справа налево с шагом отличным от 1, используя цикл с параметром

```
for (int I=n-1;I>=0;I-=step){обработка a[I];}
```

# Обработка массивов по 2 элемента

1) Элементы массива можно обрабатывать по два элемента, двигаясь с обеих сторон массива к его середине:

```
int I=0, J=N-1;  
while( I<J)  
{обработка a[I] и a[J];I++;J--;}
```

2) Элементы массива можно обрабатывать по два элемента, двигаясь от начала к концу с шагом 2 (т. е. обрабатываются пары элементов a[1]и a[2], a[3]и a[4] и т. д.)

```
int I=0;  
while (I<N-1 )  
{обработка a[I] и a[I+1];  
I+=2;}
```

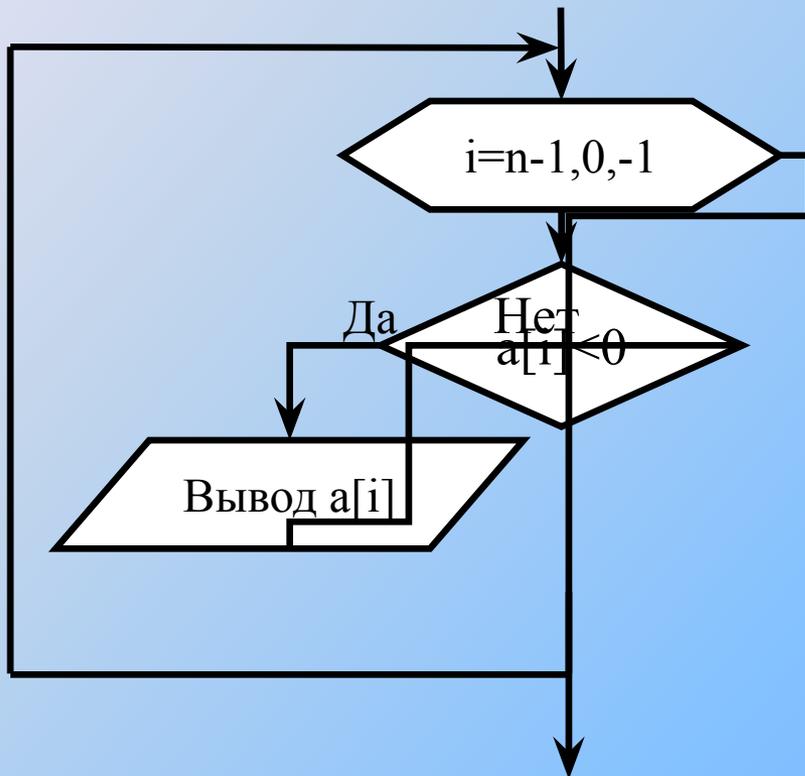
# Одномерные массивы. Пример 2.

Вывести отрицательные числа в порядке убывания индексов

Число	6	-3	-40	7	-6
Индекс	0	1	2	3	4

Алгоритм

n=5



Программный код

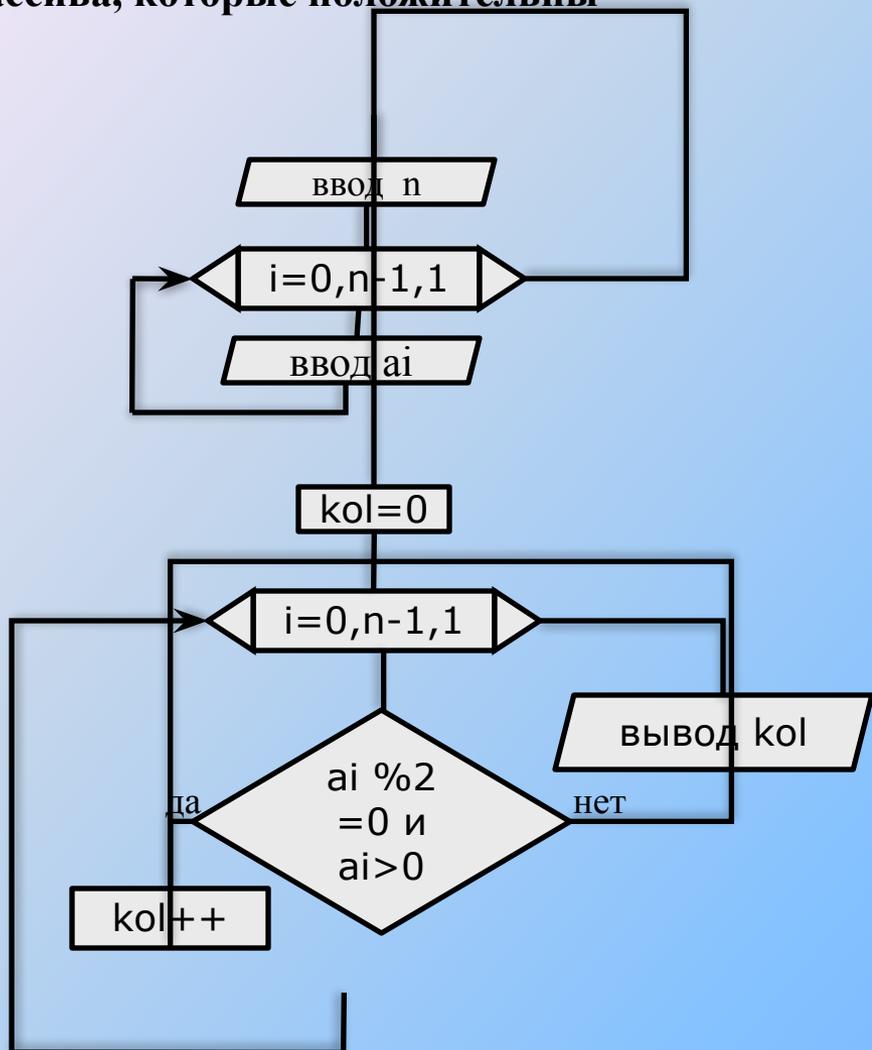
```
float a[20];  
int n,i;  
for (i=n-1; i>=0; i-- )  
    if (a[i]<0)  
        printf(“\n%d”,a[i]);
```

Результат

-6  
-40  
-3

# Одномерные массивы. Пример 3.

Сосчитать количество четных элементов массива, которые положительны



```
float a[50];
int i,n,kol;
printf("\n Input n");
scanf("%d",&n);
for(i=0; i<n; i++)
{
    printf("\n a[%d]= ",i);
    scanf("%f",&a[i]);
}
kol=0;
for(i=0;i<n;i++)
    if((a[i]%2==0)&&(a[i]>0)) kol++;
printf("kol=%d",kol);
```

Пример:

Количество элементов: 5

Элементы:

-8  
20  
1  
-3  
12

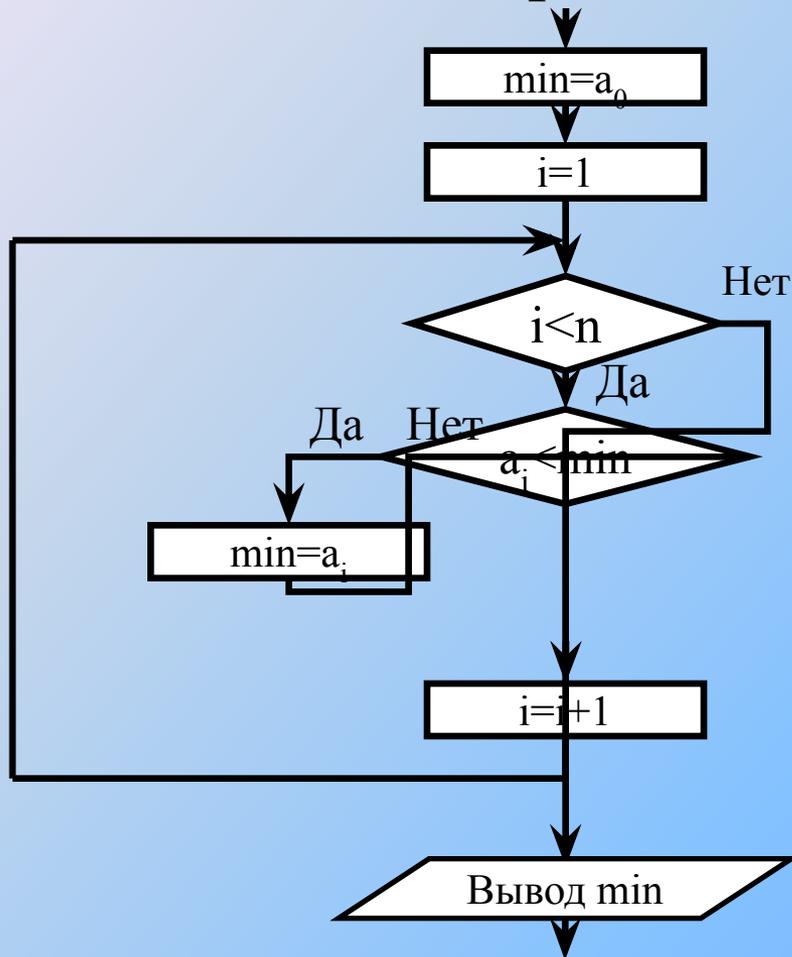
Ответ: 2

## Пример 4. Поиск минимального элемента в массиве

Число	1	5	-2	3	-6
Индекс	0	1	2	3	4

### Алгоритм

n=5



### Ход выполнения

$min = a_0(1)$

$i = 1$

$1 < 5$  – да

$a_1(5) < min(1)$  – нет

$i = 1 + 1 = 2$

$2 < 5$  – да

$a_2(-2) < min(1)$  – да

$min = a_2(-2)$

$i = 2 + 1 = 3$

$3 < 5$  – да

$a_3(3) < min(-2)$  – нет

$i = 3 + 1 = 4$

$4 < 5$  – да

$a_4(-6) < min(-2)$  – да

$min = -6$

$i = 4 + 1 = 5$

$5 < 5$  – нет

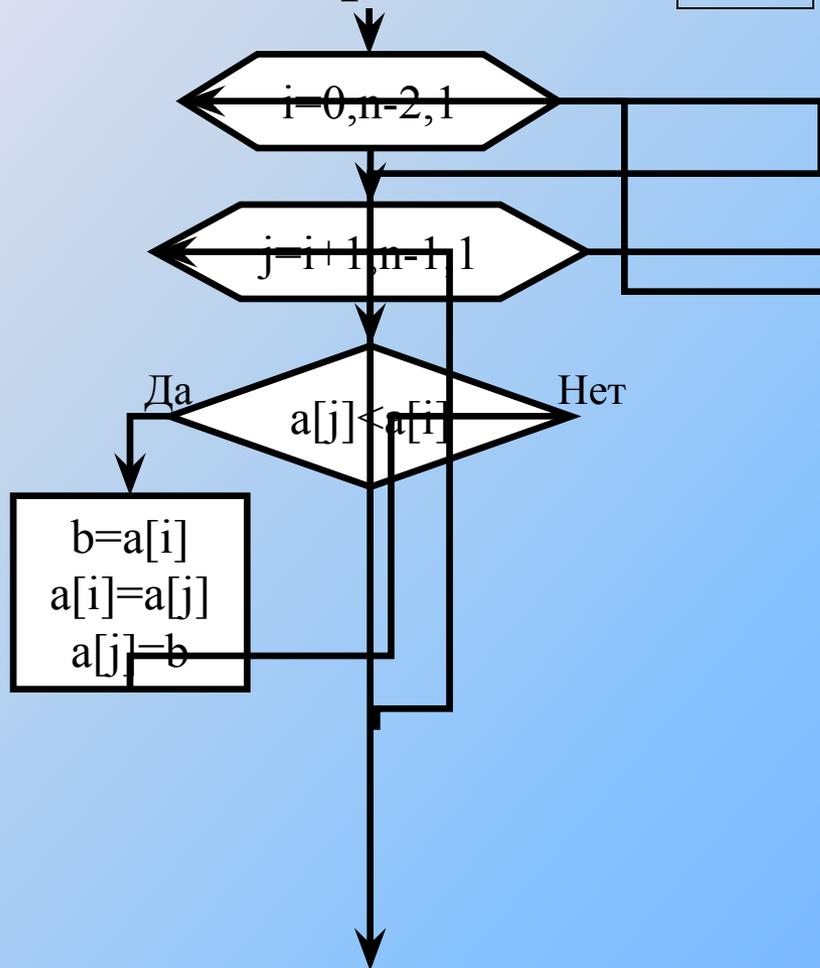
**Вывод:  $min = -6$**

# Пример 5. Сортировка массива по возрастанию элементов

Число	1	5	-2	3	-6
Индекс	0	1	2	3	4

## Алгоритм

n=5



## Ход выполнения

### Шаг 1

просмотреть все элементы, кроме  $a_0$ , сравнить с  $a_0$ , и поместить на место  $a_0$  самый маленький элемент

Результаты шага 1

Число	-6	5	1	3	-2
Индекс	0	1	2	3	4

### Шаг 2

просмотреть остальные элементы, поместить на место  $a_1$  самый маленький элемент

Результаты шага 2

Число	-6	-2	5	3	1
Индекс	0	1	2	3	4

Повторять, пока не будет проверен предпоследний элемент.

# Программа сортировки массива по возрастанию

```
#include <iostream.h>
int main(){
    const int n = 20;
        int a[n]; int i;
    for (i = 0; i<n; i++) cin >> a[i];
    for (i = 0; i<n-1; i++)
        for (int j = i + 1; j<n; j++)
            if (a[j] < a[i]) {
                int b = a[i]; a[i] = a[j]; a[j] = b;
            }
    for (i = 0; i<n; i++)cout << a[i] << ' ';
    return 0;
}
```

# Матрицы. Хранение, описание, доступ к элементам. Пример инициализации.

Многомерные массивы фиксированного размера задаются указанием каждого измерения в квадратных скобках, например, оператор

```
int matr [6][8];
```

задает описание двумерного массива из 6 строк и 8 столбцов.

Для доступа к элементу многомерного массива указываются все его индексы, например,

```
matr[i][j]=1;
```

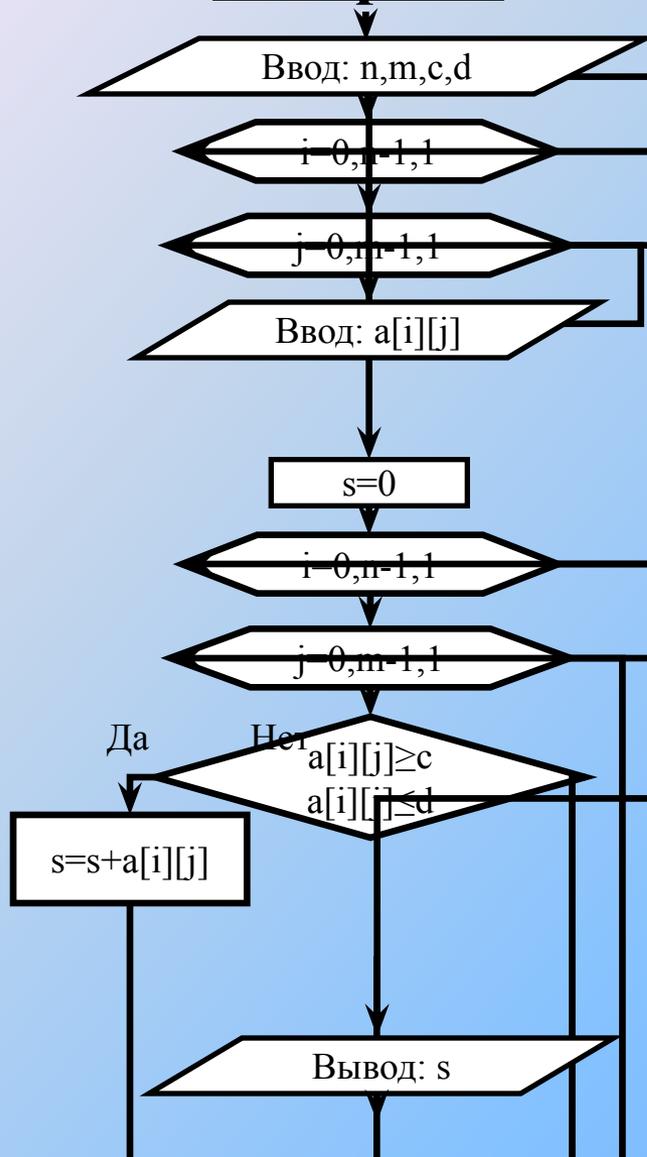
Инициализация многомерного массива:

```
int mass2 [][] = { {1, 1}, {0, 2}, {1, 0} };  
int mass2 [3][2] = {1, 1, 0, 2, 1, 0};
```

# Матрицы. Пример 2.

В матрице найти сумму чисел, принадлежащих диапазону [c,d]

## Алгоритм



## Программный код

```
int a[10][5],n,m,i,j,c,d;  
scanf("%d%d%d%d",&n,&m,&c,&d);  
for (i=0; i<n; i++)  
for (j=0; j<m; j++)  
scanf("%d",&a[i][j]);
```

```
s=0;  
for (i=0; i<n; i++)  
for (j=0; j<m; j++)  
if ((a[i][j]>=c)&&  
(a[i][j]<=d))  
s=s+a[i][j];
```

```
printf("%d",s);
```

$$A = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0m-1} \\ \vdots & \vdots & & \vdots \\ a_{n-10} & a_{n-11} & \dots & a_{n-1m-1} \end{pmatrix}$$

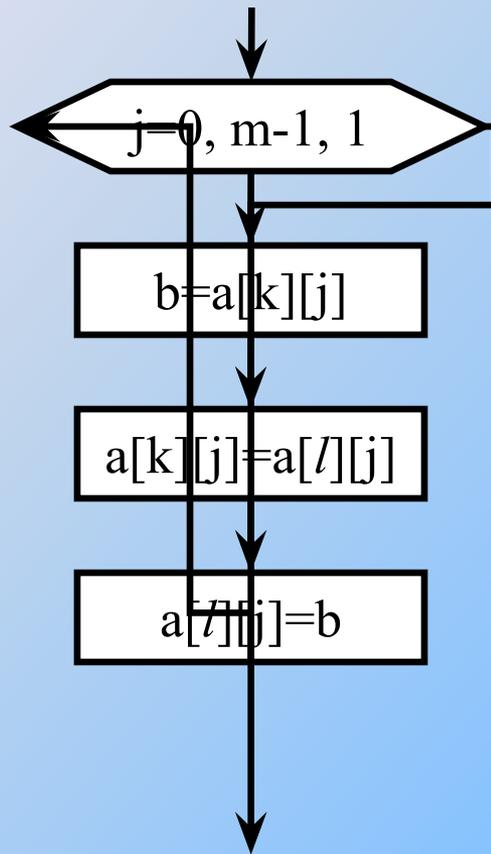
n – число строк

m – число столбцов

# Матрицы. Пример 3.

В матрице поменять местами строки с заданными номерами (k и l)

## Алгоритм



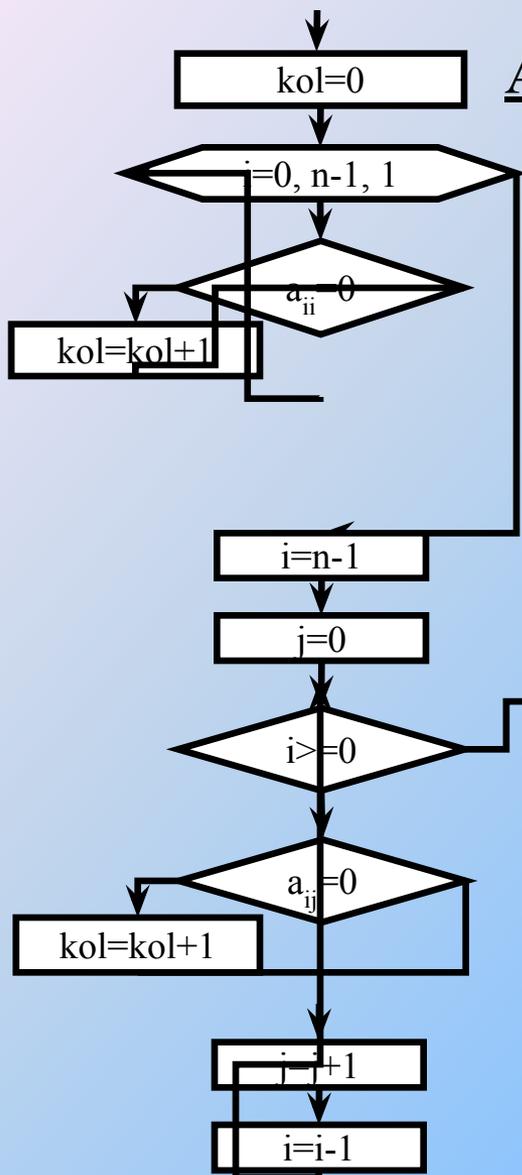
## Программный код

```
for (j=0; j<m; j++ )  
{  
    b=a[k][j];  
    a[k][j]=a[l][j];  
    a[l][j]=b;  
}
```

$$A = \begin{pmatrix} \dots & \dots & \dots & \dots & \dots \\ a_{k\ 0} & \dots & \dots & \dots & a_{k\ m-1} \\ \updownarrow & \dots & \dots & \dots & \updownarrow \\ a_{l\ 0} & \dots & \dots & \dots & a_{l\ m-1} \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

# Пример 4. Расчет количества нулей на главной и побочной диагоналях квадратной матрицы

## Алгоритм

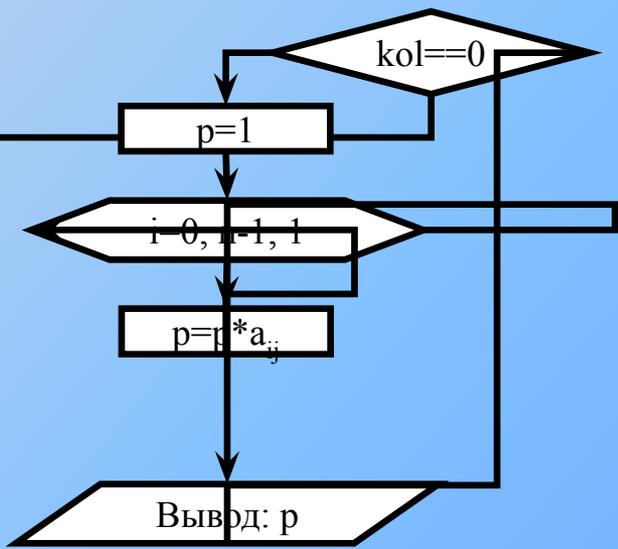


$$A = \begin{pmatrix} a_{00} & \dots & \dots & \dots & a_{0n-1} \\ \dots & a_{11} & \dots & a_{1n-2} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & a_{n-21} & \dots & a_{n-2n-2} & \dots \\ a_{n-10} & \dots & \dots & \dots & a_{n-1n-1} \end{pmatrix}$$

## Программный код

```

kol=0;
for (i=0; i<n; i++)
    if (a[i][i]==0) kol++;
for (i=n-1, j=0; i>=0; j++,i--)
    if (a[i][j]==0) kol++;
if (kol==0)
{
    p=1;
    for (i=0; i<n; i++)
        p*=a[i][j];
    printf("p=%f",p);
}
  
```



# Формирование матрицы

$$\begin{array}{l} i=0 \\ i=1 \\ i=2 \\ \dots \\ \end{array} \left[ \begin{array}{ccc} 1 & \dots & N \\ 2N & \dots & N+1 \\ 2N+1 & \dots & 3N \\ \dots & & \\ MN & \dots & (M-1)N+1 \end{array} \right]$$

```
k=M/2; z=1;
for (l=0;l<k; l++)
// столько циклов
{
    for(j=0;j<N; j++)
    {
        a[l*2][j]=z; z++;
    }
    for(j=N-1;j>=0;j--)
    {
        a[l*2+1][j]=z; z++;
    }
}
```

$$\begin{array}{l} l=0 \\ \hline l=1 \\ l=2 \end{array} \left[ \begin{array}{l} a_0 \dots \\ a_1 \dots \\ a_2 \dots \\ a_3 \dots \\ a_4 \dots \\ a_5 \dots \end{array} \right]$$

Сформировать матрицу из M строк и N столбцов. Число строк четно.