

# **Лекція 14.**

## **Операції з матрицями.**

---

Двовимірний масив цілих чисел `int a[4][4]` має чотири рядки та чотири стовпця.

<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>
<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>
<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>
<code>a[3][0]</code>	<code>a[3][1]</code>	<code>a[3][2]</code>	<code>a[3][3]</code>

У пам'яті комп'ютера масив розташовується безперервно за рядками:

`a[0][0]`, `a[0][1]`, `a[0][2]`, `a[0][3]`, `a[1][0]`, `a[1][1]`, `a[1][2]`, ... `a[3][3]`.

Двовимірні (багатовимірні) масиви оголошуються наступним чином:

```
int mas[2][5] = {1, 5, 3, 7, 4, 10, 11, 13, 14, 25 };
```

```
int mas[ ][5] = {1, 5, 3, 7, 4, 10, 11, 13, 14, 25 };
```

```
int mas[ ][5] = {{ 1, 5, 3, 7, 4 }, {10, 11, 13, 14, 25}};
```

---

## ПРИКЛАД №1

Сформууйте двовимірний масив, який складається з 5 рядків і 5 стовпців за правилом  $arr[i,j]=2ij-i$  та виведіть його на екран.

Знайдіть добуток елементів п'ятого рядка масиву.

---

1. Сформуїте двовимірний масив, який складається з 5 рядків і 5 стовпців за правилом  $arr[i,j]=2ij-i$  та виведіть його на екран.

1	<pre>const int n = 5; int arr[n][n];</pre>
2	<pre>for (int i = 0; i &lt; n; i++)     for (int j = 0; j &lt; n; j++)         arr[i][j] = 2 * i*j - i;</pre>
3	<pre>for (int i = 0; i &lt; n; i++) {     for (int j = 0; j &lt; n; j++)         printf("%4d", arr[i][j]);     printf("\n"); }</pre>

## 2. Знайдіть добуток елементів п'ятого рядка масиву.

4

```
int dob = 1;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (i==n-1) dob *= arr[i][j];
printf("\nДобуток елементів 5 рядка = %d", dob);
```

```
int dob = 1, i=n-1;
for (int j=0; j < n; j++)
    dob *= arr[i][j];
printf("\nДобуток елементів 5 рядка = %d", dob);
```

## ПРИКЛАД №2

Реалізувати програму, яка знаходить номер рядка заданого двовимірного масиву, що має максимальну за модулем суму елементів.

---

1	<pre>system("chcp 1251"); system("cls"); srand(time(NULL)); const int n = 5; const int m = 4; int arr[n][m];</pre>
2	<pre>for (int i = 0; i &lt; n; i++)     for (int j = 0; j &lt; m; j++)         arr[i][j] = rand() % 21 - 10;</pre>
3	<pre>for (int i = 0; i &lt; n; i++) {     for (int j = 0; j &lt; m; j++)         printf("%4d", arr[i][j]);     printf("\n"); }</pre>

Знайдіть номер рядка, що має максимальну за модулем суму елементів.

4

```
int summ = 0, line=0, maxSumm=0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++)
        summ += abs(arr[i][j]);
    printf("\nсумма %d рядка = %d", i, summ);
    if (summ > maxSumm) {
        maxSumm = summ;
        line = i;
    }
    summ = 0;
}
printf("\nмаксимальна сума рядок -%d", line);
```



## ПРИКЛАД №3

Дана матриця розміру  $M \times N$ .  
Поміняти місцями рядки, що  
містять мінімальний і  
максимальний елементи матриці.

---

```
system("chcp 1251"); system("cls");  
srand(time(NULL));
```

```
const int n = 5;  
const int m = 7;  
int arr[m][n];
```

```
for (int i = 0; i < m; i++)  
    for (int j = 0; j < n; j++)  
        arr[i][j] = rand() % 21 - 10;
```

```
for (int i = 0; i < m; i++) {  
    for (int j = 0; j < n; j++)  
        printf("%4d", arr[i][j]);  
    printf("\n");  
}
```

---

```
int max = arr[0][0], min = arr[0][0], maxi = 0, mini = 0;
```

```
for (int i = 0; i < m; i++)  
    for (int j = 0; j < n; j++) {  
        if (arr[i][j] < min) {  
            min = arr[i][j]; mini = i;  
        }  
        if (arr[i][j] > max) {  
            max = arr[i][j]; maxi = i;  
        }  
    }  
}
```

```
printf("min=%d mini=%d max=%d maxi=%d", min, mini, max, maxi);
```

---

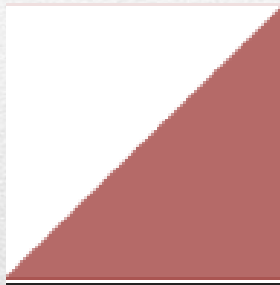
```
int tmp;
if (mini > maxi)
{
    tmp = mini;
    mini = maxi;
    maxi = tmp;
}
if (mini == maxi) printf("mini == maxi");
else {
    for (int j = 0; j < n; j++) {
        tmp = arr[mini][j];
        arr[mini][j] = arr[maxi][j];
        arr[maxi][j] = tmp;
    }
}
```

---

```
-4  7  9 -1  0
 6  0  0 -4 -8
-4 -1 -8 -4  9
-9  0  1 -6 -1
 6 -2 -6  6  6
 5 10  8 -7 -9
 2  2 -3 -7 -6
min=-9 mini=3 max=10 maxi=5
-4  7  9 -1  0
 6  0  0 -4 -8
-4 -1 -8 -4  9
 5 10  8 -7 -9
 6 -2 -6  6  6
-9  0  1 -6 -1
 2  2 -3 -7 -6
```

## ПРИКЛАД №4

Знайти суму елементів заштрихованої частини матриці



a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]

1	<pre>const int n = 4; int arr[n][n];</pre>
2	<pre>for (int i = 0; i &lt; n; i++)     for (int j = 0; j &lt; n; j++)         arr[i][j] = rand() % 11 - 5;</pre>
3	<pre>for (int i = 0; i &lt; n; i++) {     for (int j = 0; j &lt; n; j++)         printf("%4d", arr[i][j]);     printf("\n"); }</pre>
4	<pre>int summ = 0, i, j, ij=0; for (j = n - 1; j &gt;= 0; j--) {     for (i = ij; i &lt; n; i++) {         summ += arr[i][j];         printf("\narr[%d][%d]=%d, summ=%d", i, j, arr[i][j], summ);     }     ij++; } printf("\ncymma= %d", summ);</pre>

```

-3   2   -2   0
 2   1   -2   0
 1  -4   -3   2
-3  -3   4   1

```

```

arr[0][3] = 0, summ=0
arr[1][3] = 0, summ=0
arr[2][3] = 2, summ=2
arr[3][3] = 1, summ=3
arr[1][2] = -2, summ=1
arr[2][2] = -3, summ=-2
arr[3][2] = 4, summ=2
arr[2][1] = -4, summ=-2
arr[3][1] = -3, summ=-5
arr[3][0] = -3, summ=-8

```

сумма= -8

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]

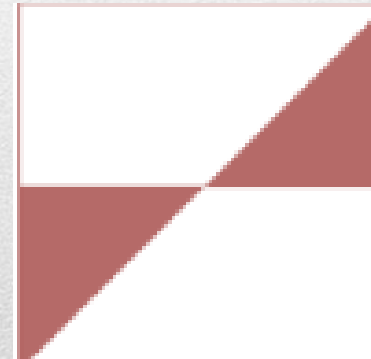


## ПРИКЛАД №5

Знайти суму елементів заштрихованої частини матриці



a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]



```
int summ = 0, ij = 0;
for (int j = n - 1; j >= 0; j--) {
    for (int i = ij; i < n/2; i++) {
        summ += arr[i][j];
        printf("\narr[%d][%d]=%d, cymma =%d", i, j, arr[i][j], summ);
    }
    ij++;
}

ij = n-1;
for (int j =0; j <n/2; j++) {
    for (int i = ij; i >=n / 2; i--) {
        summ += arr[i][j];
        printf("\narr[%d][%d]=%d, cymma =%d", i, j, arr[i][j], summ);
    }
    ij--;
}
printf("\ncymma= %d", summ);
```

---

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]
a[3][0]	a[3][1]	a[3][2]	a[3][3]

8	0	3	-6
1	-10	-6	-7
1	6	-5	-2
-3	-6	-5	7

arr[0][3]=-6, сумма =-6  
arr[1][3]=-7, сумма =-13  
arr[1][2]=-6, сумма =-19  
arr[3][0]=-3, сумма =-22  
arr[2][0]=1, сумма =-21  
arr[2][1]=6, сумма =-15  
сумма= -15

a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]	a[0][5]
a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	a[1][5]
a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]	a[2][5]
a[3][0]	a[3][1]	a[3][2]	a[3][3]	a[3][4]	a[3][5]
a[4][0]	a[4][1]	a[4][2]	a[4][3]	a[4][4]	a[4][5]
a[5][0]	a[5][1]	a[5][2]	a[5][3]	a[5][4]	a[5][5]

```

8   1   9   1  -4   6
3  -3   8  -6  10  -7
-9   8   3   4   5   9
8   -2  10   9   4   6
-4  -6  -2  -2   6   8
-9   6  -5  -4  -6  10

```

```

arr[0][5]=6, summ=6
arr[1][5]=-7, summ=-1
arr[2][5]=9, summ=8
arr[1][4]=10, summ=18
arr[2][4]=5, summ=23
arr[2][3]=4, summ=27
arr[5][0]=-9, summ=18
arr[4][0]=-4, summ=14
arr[3][0]=8, summ=22
arr[4][1]=-6, summ=16
arr[3][1]=-2, summ=14
arr[3][2]=10, summ=24
сумма= 24

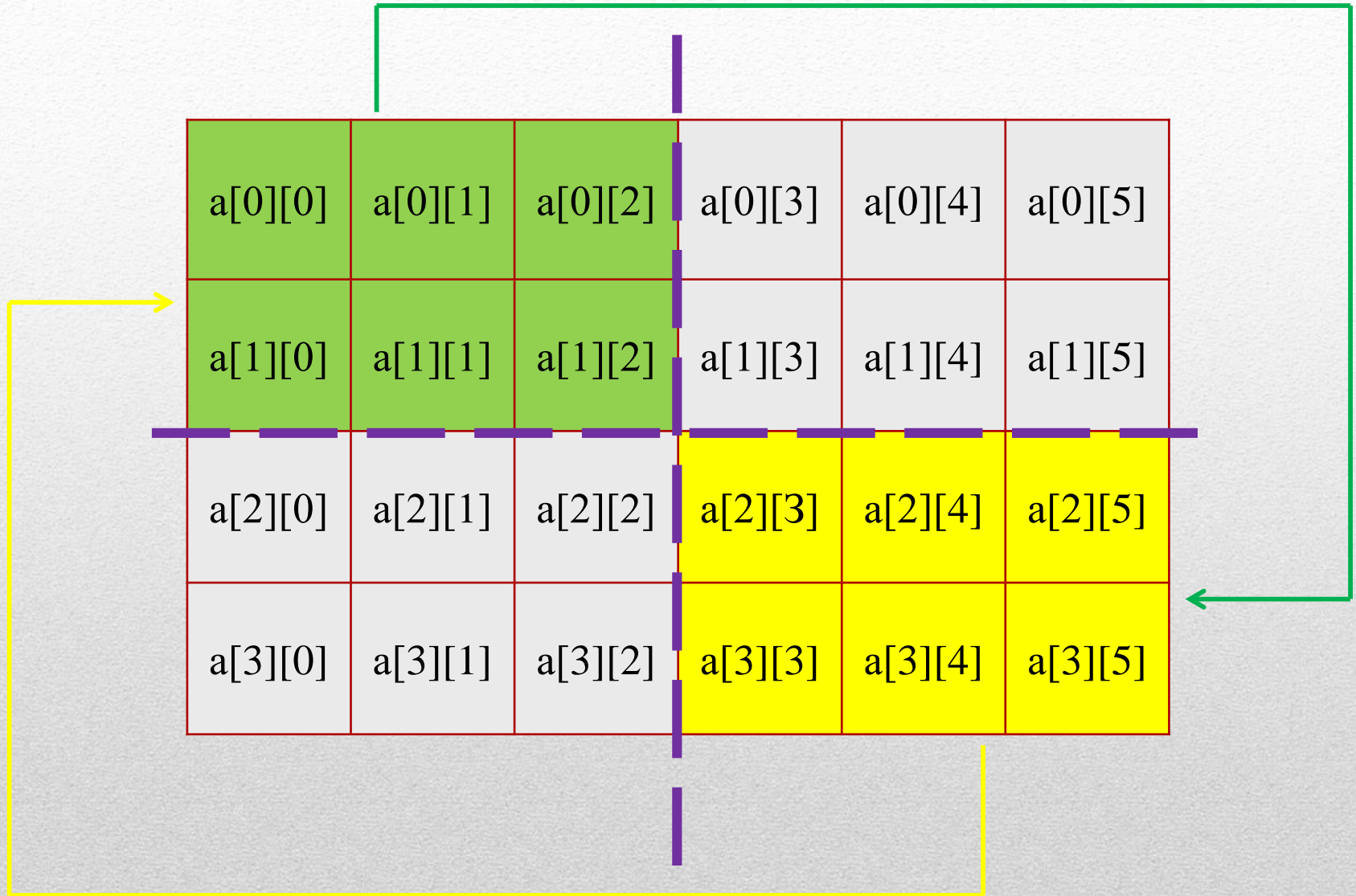
```

## ПРИКЛАД №6



Дана матриця розміру  $M \times N$  ( $M$  і  $N$  - парні числа). Поміняти місцями ліву верхню і праву нижню чверті матриці.

---



```

const int n = 6; const int m = 4;
int arr[m][n];
printf("\n");
for (int i = 0; i < m; i++)
    for (int j = 0; j < n; j++)
        arr[i][j] = rand() % 20 - 10;

for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
        printf("%4d", arr[i][j]);
    printf("\n");
}

int tmp;
for (int i = 0, ii=m/2; i < m/2; i++,ii++)
    for (int j = 0, jj = n / 2; j < n / 2; j++,jj++) {
        tmp = arr[i][j];
        arr[i][j] = arr[ii][jj];
        arr[ii][jj]=tmp;
    }

printf("\n-----\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++)
        printf("%4d", arr[i][j]);
    printf("\n");
}

```

$$\begin{array}{ccc|ccc} -8 & 8 & -3 & 4 & -9 & -5 \\ -6 & 7 & -4 & 1 & 4 & 0 \\ \hline -8 & -7 & -4 & 4 & 6 & 2 \\ 5 & -4 & -5 & -6 & -10 & -8 \end{array}$$

---

$$\begin{array}{ccc|ccc} 4 & 6 & 2 & 4 & -9 & -5 \\ -6 & -10 & -8 & 1 & 4 & 0 \\ \hline -8 & -7 & -4 & -8 & 8 & -3 \\ 5 & -4 & -5 & -6 & 7 & -4 \end{array}$$