

# Многомерни масиви

## 1. Вход и изход на матрица

Напишете програма, която въвежда брой **редове** и брой **колони**. След което въвежда елементите на двумерен масив (матрица) със съответния брой редове и колони. Всички елементи на масива ще са цели числа. Изведете получения двумерен масив

### Примери

Вход	Изход	Вход	Изход
2	1 2 3 5	3	1 2 3
4	8 6 9 4	3	9 8 7
1		1	4 5 6
2		2	
3		3	
5		9	
8		8	
6		7	
9		4	
4		5	
		6	

### Решение

Може да си помогнете със следния код:

```
int rows = int.Parse(Console.ReadLine());
int cols = int.Parse(Console.ReadLine());
int[,] matrix = new int[rows, cols];

for (int row = 0; row < rows; row++)
{
    for (int col = 0; col < cols; col++)
    {
        matrix[row, col] = int.Parse(Console.ReadLine());
    }
}

for (int row = 0; row < rows; row++)
{
    for (int col = 0; col < cols; col++)
    {
        Console.Write(matrix[row, col] + " ");
    }
    Console.WriteLine();
}
```

## 2. Средноаритметично по редове

Напишете програма, която въвежда брой **редове** и брой **колони**. След което въвежда елементите на двумерен масив (матрица) със съответния брой редове и колони. Всички елементи на масива ще са цели числа. Изведете двумерния масив, като на всеки ред прибавите по 1 елемент в края му, който да бъде

равен на средноаритметичното от всички елементи в съответния ред. При извеждане на масива го форматирайте, така че всеки елемент да заема 10 позиции

## Примери

Вход	Изход				
2	1	2	3	5	2.75
4	8	6	9	4	6.75
1					
2					
3					
5					
8					
6					
9					
4					

## Подсказки

- Когато отпечатвате елемента си, използвайте нещо подобно:  

```
Console.Write("{0, 10}", matrix[row, col]);
```
- Със започването на всеки ред задавайте стойността на променливата, в която пазите сумата на елементите му на 0. Когато приключвате с обхождането на реда, изчислете средноаритметичното му и го изведете.

## 3. Минимум по колони

Напишете програма, която въвежда брой **редове** и брой **колони**. След което въвежда елементите на двумерен масив (матрица) със съответния брой редове и колони – елементите на всеки ред от масива ще са на отделен ред. Всички елементи на масива ще са цели числа. Изведете двумерния масив, като накрая добавите един нов ред – всеки елемент в този ред показва минималния елемент от колоната, която стои над него. При извеждане на масива го форматирайте, така че всеки елемент да заема 5 позиции

## Примери

Вход	Изход				
3	1	2	3	5	
4	8	6	9	4	
1 2 3 5	5	8	4	3	
8 6 9 4	1	2	3	3	
5 8 4 3					

## Подсказки

- Когато въвеждате матрицата, създайте си помощен масив за реда. Въвеждането му от един ред е аналогично на списъците:

```

for(int row = 0; row < rows; row++) {
    int[] rowArray = Console.ReadLine().Split(' ').Select(int.Parse).ToArray();
    for(int col = 0; col < cols; col++) {
        matrix[row, col] = rowArray[col];
    }
}

```

- Създайте едномерен масив, който ще съхранява минималните елементи от всяка колона
- Сменете местата на редовете и колоните в обхождането, така че външния цикъл да отговаря за колоните, а вътрешния за редовете.
- След пълното извъртане на вътрешния цикъл, съхранете намерения минимален елемент в едномерния масив по подобен начин: **minElements[col] = min;**
- Отпечатайте матрицата, като непосредствено след нея, отпечатайте и едномерния масив – използвайте едно и също форматиране.

## 4. Лотариен Билет

Прасчо си купил лотариен билет. Тъй като Прасчо не разбирал много-много, но пък имал голям късмет, отишъл при Мечо Пух да му помогне с „разшифрирането“ на лотарийния билет. Лотарийния билет след изтъркване представлява табличка от числа с **n реда** и **m колони**. Един билет печели, ако:

- Сумата от елементите намиращи се на главния диагонал е равна на сумата от елементите намиращи се на вторичния диагонал
- Сумата от елементите **НАД** главния диагонал е **четна**
- Сумата от елементите **ПОД** главния диагонал е **нечетна**

Точната печалба се определя като средноаритметично от следните суми:

- Сума от **всички** елементи намиращи се **ПОД** главния диагонал
- Сумата на елементите, които са **четни** числа и се намират **точно** на главния диагонал
- Сумата на **четните** по стойност елементи, които са на външни редове (т.е. първи и последен)
- Сумата на **нечетните** по стойност елементи, които са на външни колони (т.е. първа и последна)

Нормално и напълно очаквано е едно и също число да принадлежи към повече от една от тези суми. Всички числа в таблицата са **положителни цели числа**.

От вас се очаква да изведете: YES, ако билетът печели, както и печалбата му, закръглена до втори знак след запетаята и NO, в противен случай.

## Вход

Размерностите на таблицата ще бъдат въведени от един и същи ред, разделени с интервал.

Таблицата ще бъде въведена по редове, като всеки елемент на даден ред е разделен с интервал.

## Примери

Вход	Изход	Обяснение
3 3 1 2 2 3 5 6 8 8 9	YES The amount of money won is: 13.00	Сумата от главния диагонал е 15, от вторичния също. Сумата на елементите над главния диагонал е 10 (2+2+6), сумата на елементите под главния диагонал е 19 (3+8+8), следователно

		<p>билета изпълнява условията да е печеливш.</p> <p>Оттам нататък изчисляваме печалбата:</p> <ul style="list-style-type: none"> <li>• Сумата под диагонала е 19 (3+8+8)</li> <li>• Сумата от четните елементи точно на главния диагонал е 0 (на диагонала НЕ лежат четни елементи)</li> <li>• На външните редове лежат следните четни числа: 2+2+8+8=20</li> <li>• На външните колони лежат следните нечетни: 1+3+9=13</li> </ul> <p>Тяхното средно аритметично е <math>(19+0+20+13)/4 = 13.00</math></p>
--	--	---

## Подсказки

- Въведете масива, след което го обходете. Правете следните проверки:
  - Ако номерът на реда съвпада с номера на колоната, то елемента с тези индекси лежи **НА** главния диагонал
  - Ако номерът на реда е по-малък от номера на колоната, то елемента с тези индекси лежи **НАД** главния диагонал
  - Ако номерът на реда е по-голям от номера на колоната, то елемента с тези индекси лежи **ПОД** главния диагонал
  - Ако сборът от номерата на реда и колоната е равен на броя на редовете – 1, то елемента с тези индекси лежи **НА** вторичния диагонал.
  - Проверка за четен елемент може да извършите по подобен начин:  
`lotteryTicket[row, col]%2==0`
  - Проверка за четен елемент може да извършите по подобен начин:  
`lotteryTicket[row, col]%2==1`