

שכנים (row,col)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1) num	(1,2)
(2,0)	(2,1)	(2,2)

שאלה 14

הגדרה:

- מסגרת המערך הדו-ממדי מורכבת מתאים הנמצאים בשורה 0, עמודה 0, שורה אחרונה ועמודה אחרונה.
- "תא פנימי" במערך דו-ממדי הוא איבר שלא נמצא במסגרת המערך.
- תא פנימי במערך נקרא "עצוב" אם אין אף שכן (מתוך שמונת השכנים) שערכו זהה לערך התא. ערך שנימצא ב"תא עצוב" נקרא "ערך עצוב".

לדוגמא:

- תא (1,1) הוא "תא עצוב" כי ערכים של כל שכנים שלו שונים מערכו. שכנים של תא (1,1) מסומנים באפור. ערך 10 הוא "ערך עצוב".
- גם תא (3,2) הוא תא "עצוב", ערך 21 הוא "ערך עצוב".
- גם תא (1,4) הוא תא "עצוב", ערך 123 הוא ערך עצוב.
- תא (2,3) או תא (3,3) הם לא "עצובים".

	0	1	2	3	4	5
0	5	4	1	0	0	3
1	1	10	7	1	123	3
2	6	1	7	0	0	2
3	1	10	21	5	5	2
4	1	10	1	10	10	1

(6 נק') א. כתבו פעולה אשר מקבלת מערך דו-ממדי arr ומספר שלם num. הפעולה תחזיר true אם קיים במערך

arr "תא עצוב" שהערך שלו num, ולא הפעולה תחזיר false.

(6 נק') ב. כתבו פעולה אשר מקבלת מערך דו-ממדי arr ומחזירה מערך חד-ממדי הכולל את כל ערכים

ה"עצובים". שימו לב שבמערך החדש כל ערך צריך להופיע פעם אחת בלבד. אם במערך arr אין תאים

"עצובים", יש להחזיר null.

(6 נק') ג. כתבו פעולה אשר מקבלת מערך דו-ממדי של מספרים שלמים. הפעולה תדפיס את **המספר הדו-**

ספרתי החיובי הגדול ביותר שעבורו קיים במערך "תא עצוב". אם אין במערך "תא עצוב" שערכו הוא


מספר דו-ספרתי חיובי, תודפס הודעה מילולית מתאימה.

עבור דוגמה הנ"ל הפעולה תדפיס 21.

סעיף א'

```
static public bool sadCell(int[,] arr, int num)
{
    for (int row = 1; row < arr.GetLength(0) - 1; row++)
    {
        for (int col = 1; col < arr.GetLength(1) - 1; col++)
        {
            if (num == arr[row, col])
            {
                if(sad(arr, row, col))
                {
                    return true;
                }
            }
        }
    }
    return false;
}
```

פונקציית עזר



```
static public bool sad(int[,] arr, int r, int c)
{
    for(int row = r - 1; row <= r + 1; row++)
    {
        for(int col = c - 1; col <= c + 1; col++)
        {
            if (!((r == row) && (c == col)) && (arr[row, col] == arr[r, c]))
            {
                return false;
            }
        }
    }
    return true;
}
```

הסבר בקישור:

<https://youtu.be/-3vIZTUcWLU>

סעיף ב'

```
static public int[] newArray(int[,] arr)
{
    int[] Array = new int[(arr.GetLength(0) - 2) * (arr.GetLength(1) - 2)];
    int index = 0;

    for (int row = 1; row < arr.GetLength(0) - 1; row++)
    {
        for (int col = 1; col < arr.GetLength(1) - 1; col++)
        {
            if (sadCell(arr, arr[row, col]))
            {
                if (!anExistingNumber(Array, arr[row, col], index))
                {
                    Array[index++] = arr[row, col];
                }
            }
        }
    }

    if (index == 0)
    {
        return null;
    }

    int[] newArray = new int[index];
    for (int k = 0; k < newArray.Length; k++)
    {
        newArray[k] = Array[k];
    }

    return newArray;
}
```

פונקציית עזר



```
static public bool anExistingNumber(int[] a, int num, int L)
{
    for (int k = 0; k < L; k++)
    {
        if (a[k] == num)
        {
            return true;
        }
    }

    return false;
}
```

הסבר בקישור:

<https://youtu.be/UN7LVpU0830>

סעיף ג'

```
static public void bigNum(int[,] arr)
{
    int bigger = 0;

    for (int row = 1; row < arr.GetLength(0) - 1; row++)
    {
        for (int col = 1; col < arr.GetLength(1) - 1; col++)
        {
            if (((arr[row, col] > 9) && (arr[row, col] < 100)) && (arr[row, col] > bigger))
            {
                if (sadCell(arr, arr[row, col]))
                {
                    bigger = arr[row, col];
                }
            }
        }
    }

    if (bigger != 0)
    {
        Console.WriteLine(bigger);
    }
    else
    {
        Console.WriteLine("There is no sad cell with a double digit number");
    }
}
```

הסבר הקישור:

<https://youtu.be/qePo1ZBgJdk>