

לתרגול אדפטיבי, בדיקת תשובות אוטומטית, ועוזר בינה מלאכותית חינם, היכנסו ל-www.TIRGUL.ai

יצירת מערך

מערך הוא שם אחד שמצביע על כמה תאים מאותו סוג. ב-`new int[5]` נוצרים 5 תאים, ובכולם 0. אפשר גם לאתחל ברשימת ערכים בסוגריים מסולסלים. הגודל יכול להיות משתנה שנקלט: `new int[n]`.

```
int[] grades = new int[5];
int[] arr = {9, 4, 7, 1, 3};
string[] names = {"Dana", "Omer", "Noa"};
```

גישה לתא ו-Length

המיקומים מתחילים ב-0. `arr.Length` הוא מספר התאים, ולכן המיקום האחרון הוא `Length - 1`. תא במערך מתנהג כמו משתנה רגיל.

```
int[] arr = {9, 4, 7, 1, 3};
arr[0] = 10;
int x = arr[1];
Console.WriteLine(arr[0] + x);
Console.WriteLine(arr[arr.Length - 1]);
```

הפלט:

```
14
3
```

מעבר על מערך בלולאה

מילוי מקלט, ואחריו סריקה. מונה, סוכם, מקסימום ומינימום עובדים כמו בנושא 7, כש-`arr[i]` מחליף את הקלט. למקסימום ולמינימום נוח לאתחל לפי `arr[0]`.

```
int[] arr = new int[5];
for (int i = 0; i < arr.Length; i++)
{
    arr[i] = int.Parse(Console.ReadLine());
}
int sum = 0;
for (int i = 0; i < arr.Length; i++)
{
    sum += arr[i];
}
```

מערך מונים

כל תא במערך הוא מונה נפרד, והערך עצמו קובע איזה תא גדל. לערכים מ-1 עד 6 יוצרים 7 תאים, כדי שיהיה מיקום 6.

```
int[] counters = new int[7];
for (int i = 0; i < rolls.Length; i++)
{
    int r = rolls[i];
    counters[r] = counters[r] + 1;
}
```

פעולה שמקבלת מערך

בכותרת כותבים `int[]` ושם. בקריאה שולחים רק את שם המערך, בלי סוגריים מרובעים: `SumArray(arr)`.

```
static int SumArray(int[] array1)
{
    int sum = 0;
    for (int i = 0; i < array1.Length; i++)
    {
        sum += array1[i];
    }
    return sum;
}
```

מערך שנשלח לפעולה אינו עותק. פעולה שמשנה תא במערך משנה את המערך המקורי, גם בלי `return`.

העתקת מערך

`int[] arr2 = arr;` לא יוצרת מערך חדש, אלא מצביע נוסף על אותו מערך. שינוי דרך `arr2` נראה גם דרך `arr`. להעתקה אמיתית יוצרים מערך חדש ב-`new` ומעתיקים תא אחרי תא.

```
int[] copy = new int[arr.Length];
for (int i = 0; i < arr.Length; i++)
{
    copy[i] = arr[i];
}
```

העתקה בסדר הפוך:

```
.copy[i] = arr[arr.Length - 1 - i];
```

מערך דו-ממדי: יצירה וגישה לתא

טבלה של שורות ועמודות. ב- `new int[3, 4]` יש 3 שורות ו-4 עמודות. בגישה לתא כותבים קודם את מספר השורה ואז את מספר העמודה, ושניהם מתחילים ב-0. `GetLength(0)` מחזירה את מספר השורות, `GetLength(1)` - את מספר העמודות.

```
int[,] m = new int[3, 4];
int[,] a = {{2, 4, 6}, {1, 3, 5}};
m[2, 1] = 10;
Console.WriteLine(a[1, 2]);
Console.WriteLine(m.GetLength(0));
Console.WriteLine(m.GetLength(1));
```

הפלט:

```
5
3
4
```

מערך דו-ממדי: מעבר על כל התאים

לולאה חיצונית על השורות ולולאה פנימית על העמודות. למילוי מקלט כותבים בפנים

```
.m[i, j] = int.Parse(Console.ReadLine());
```

```
for (int i = 0; i < m.GetLength(0); i++)
{
    for (int j = 0; j < m.GetLength(1); j++)
    {
        Console.Write(m[i, j] + " ");
    }
    Console.WriteLine();
}
```

מערך דו-ממדי: שורה, עמודה ואלכסונים

לשורה אחת או לעמודה אחת מספיקה לולאה אחת: בשורה `r` עוברים על `m[r, j]`, ובעמודה `c` על `m[i, c]`. אלכסונים יש רק במערך ריבועי: באלכסון הראשי התא הוא `m[i, i]`, ובאלכסון המשני `m[i, n - 1 - i]`.

```
int n = m.GetLength(0);
int sum1 = 0;
int sum2 = 0;
for (int i = 0; i < n; i++)
{
    sum1 += m[i, i];
    sum2 += m[i, n - 1 - i];
}
```