

## רקורסיה במחשנית

## תרגיל 1

```
public static int What (Stack<int> stk)
{
    if (stk.IsEmpty())
        return 0;
    int x = stk.Pop();
    int sod = What (stk);
    stk.Push (x);
    return sod + 1;
}
```

טענת כניסה: stk מחשנית של מספרים שלמים.

עקוב אחר ביצוע הפעולה בטבלת מעקב ורשום את

טענת היציאה של הפעולה.

דוגמת מעקב: עבור המחשנית [3, 9, 12, 5, 4, 6] stk:

המחשנית בזימון	מחשנית ריקה?	x	ערך מוחזר sod	המחשנית בחזרה
[3, 9, 12, 5, 4, 6]	לא	3	$\underline{\underline{5}} + 1 = 6$	[3, 9, 12, 5, 4, 6]
[9, 12, 5, 4, 6]	לא	9	$\underline{\underline{4}} + 1 = 5$	[9, 12, 5, 4, 6]
[12, 5, 4, 6]	לא	12	$\underline{\underline{3}} + 1 = 4$	[12, 5, 4, 6]
[5, 4, 6]	לא	5	$\underline{\underline{2}} + 1 = 3$	[5, 4, 6]
[4, 6]	לא	4	$\underline{\underline{1}} + 1 = 2$	[4, 6]
[6]	לא	6	$\underline{\underline{0}} + 1 = 1$	[6]
[ ]	כן		0	[ ]

הערך המוחזר: 6  
 הפעולה סופרת ומחזירה את מספר האיברים במחשנית.  
**שימו לב!** בסיום המחשנית נשארת ללא שינוי

## תרגיל 2

לפניכם פעולה רקורסיבית:

```
public static void What (Stack<int> s1, Stack<int> s2)
{
    if (!s1.IsEmpty())
    {
        int x = s1.Pop();
        What (s1, s2);
        s1.Push(x);
        s2.Push(x);
    }
}
```

טענת כניסה: s2 מחשנית ריקה.

עקוב אחר ביצוע הפעולה בטבלת מעקב ורשום את

טענת היציאה של הפעולה.

## תרגיל 3

לפניכם פעולה רקורסיבית:

```

public static bool Mystery (Stack<int> stk, int x)
{
    if (stk.IsEmpty())
        return false;
    int y = stk.Pop();
    if (x == y)
    {
        stk.Push (y);
        return true;
    }
    bool sod = Mystery (stk, x);
    stk.Push (y);
    return sod;
}

```

טענת כניסה:  $stk$  מחסנית של מספרים שלמים.  $x$  מספר שלם.

עקוב אחר ביצוע הפעולה בטבלת מעקב ורשום את טענת היציאה של הפעולה.

## תרגיל 4

לפניכם פעולה רקורסיבית:

```

public static bool Mystery (Stack<int> s1, Stack<int> s2)
{
    if (s1.IsEmpty() && s2.IsEmpty())
        return true;
    if (s1.IsEmpty() || s2.IsEmpty())
        return false;
    int x = s1.Pop();
    int y = s2.Pop();
    bool sod = (x == y) && Mystery (s1, s2);
    s1.Push (x);
    s2.Push (y);
    return sod;
}

```

טענת כניסה:  $s1$  ו- $s2$  מחסניות של מספרים שלמים

עקוב אחר ביצוע הפעולה בטבלת מעקב ורשום את טענת היציאה של הפעולה.

**bool sod = (x == y) && Mystery (s1, s2);**

שים לב, זהו ביטוי לוגי.

תוצאת הביטוי הלוגי תושם במשתנה בוליאני sod

בטבלת המעקב רשום עמודה עבור  $x == y$ ,

עמודה עבור הערך המוחזר מהזימון הרקורסיבי

ועמודה עבור sod שיכיל את התוצאה של שתי העמודות הקודמות

## תרגיל 5

לפניכם פעולה רקורסיבית המשתמשת בפעולה sod :

```
public static int What (Stack<int> stk)
{
    if (stk.IsEmpty())
        return 0;
    int x = stk.Pop();
    int whoAmI = Sod (x, (What (stk)));
    stk.Push (x);
    return whoAmI;
}
```

טענת כניסה : stk מחסנית של מספרים שלמים  
עקוב אחר ביצוע הפעולה בטבלת מעקב ורשום את  
טענת היציאה של הפעולה.

```
public static int Sod (int x, int y)
{
    if (x > y)
        return x;
    return y;
}
```

## תרגיל 6

```
public static int What (Stack<int> stk)
{
    if (! stk.IsEmpty())
    {
        int x = stk.Pop();
        if (stk.IsEmpty())
            return x;
        int b = What (stk);
        stk.Push (x);
        return b;
    }
    return -1;
}
```

(המשך התרגיל בעמוד הבא)

```
public static bool Mystery (Stack<int> stk, int t)
{
    if (t > 0)
    {
        int a = What (stk);
        if (a == -1)
            return true;
        int b = stk.Pop();
        if (a != b)
        {
            stk.Push (b);
            return false;
        }
        return Mystery (stk, t-1);
    }
    return true;
}
```

- א. נתונה המחסנית הבאה :  $stk: [12, 11, 10, 9, 8, 10, 11, 12]$  .  
עקוב אחר הפעולה what והמחסנית  $stk$  ורשום את טענת היציאה. התייחס למצב המחסנית בסיום הפעולה.
- ב. עקוב אחר משפט הזימון :  $mystery (stk, 3)$  ורשום מה יוחזר. (אין צורך להראות מעקב נוסף אחר הפעולה What).
- ג. מה יחזיר הזימון :  $mystery (stk, 4)$  . (אין צורך להראות מעקב).
- ד. רשום את טענת היציאה של הפעולה  $Mystery (stk, t)$  . התייחס ל-  $t$  ולמצב המחסנית בסיום הפעולה.