

## עוד תרמיים הרקורסיה ...

1. א. כתוב פעולה רקורסיבית המקבלת שני מספרים שלמים ומחזירה את המנה שלהם בעזרת פעולות חיבור וחסור בלבד.
- ב. מה היית משנה בפעולה שכתבת בסעיף הקודם, על מנת שהיא תחזיר את השארית במקום את המנה.
2. כתוב פעולה רקורסיבית המקבלת מספר שלם  $n$ , ומחזירה את ערך הביטוי הבא:  
 $2 * n$  או  $1 + 2^2 + 2 * 3 + 4^2 + 2 * 5 + \dots + n^2$
3. כתוב פעולה רקורסיבית המקבלת מערך, ומחזירה את מספר האיברים החיוביים במערך.
4. כתוב פעולה רקורסיבית שתדפיס את לוח הכפל. הפעולה תקבל 2 פרמטרים שיצינו את הערך הנוכחי של זוג האינדקסים (הקריאה לפעולה תהיה עם הערכים 1,1).
5. נתונה הפעולה multiply המקבלת שני מספרים ואמורה להחזיר את תוצאת הביטוי  $x * y$  (נתון  $x > 0$ ).

```
public static int multiply (int x, int y)
{
    if ( x == 1 )
        ( הוראה 1 );
    else
        ( הוראה 2 );
}
```

איזה צמד של הוראות ישלים את הפעולה כך שיוחזר הערך המבוקש ?

	הוראה 1	הוראה 2
1	<code>return x * y</code>	<code>return multiply (x-1, y+1)</code>
2	<code>return y</code>	<code>return multiply (x-1, y+y)</code>
3	<code>return y</code>	<code>return multiply (x-1, y) + y</code>
4	<code>return y</code>	<code>return multiply (x-1, y+y)</code>
5	<code>return y</code>	<code>return multiply (x-1, y) * y</code>

6. פאלינדרום - מילה שקריאתה מימין זהה לקריאתה משמאל.  
 דוגמא: 'סוס', 'loppol', 'abba'.
- א. כתוב פעולה רקורסיבית המקבלת מחרוזת כפרמטר ומחזירה "אמת" אם המחרוזת פאלינדרום, ו-"שקר" אחרת.
- ב. כתוב תכנית הקולטת מחרוזת ומדפיסה הודעה מתאימה - האם המחרוזת פאלינדרום.

7. לפניך מספר סדרות. עבור כל סדרה, קבע האם היא יכולה להיות מודפסת בעזרת הליך רקורסיבי. נמק!

- (1) 1, 1, 2, 2, 3, 3, 4, 4, 3, 3, 2, 2, 1, 1
- (2) 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 2, 2, 1
- (3) 9, 7, 5, 3, 1, 11, 13, 1, 17, 19
- (4) 1, 2, 3, 1, 2, 3, 1, 2, 3
- (5) 1, 2, 3, 7, 7, 7, 1, 2, 3
- (6) 1, 2, 3, 4, 5, 6, 7, 6, 4, 2

8. מה מבצעת התכנית הבאה ?

```
import java.util.Scanner;
class What{
    public static void hafoch (int j, int n )
    {
        if (j == n) System.out.print ( n + " ");
        else
        {
            System.out.print ( j + " ");
            hafoch ( j + 1, n);
            System.out.print ( j + " ");
        }
    }

    public static void main (String [] args)
    {
        int i, k, size ;
        Scanner input = new Scanner (System.in);

        System.out.print ("Enter a number (between 4 to 9) → ");
        size = input.nextInt() ;

        i = 1 ;
        hafoch (i, size) ;
        System.out.println () ;
    }
}
```

9. כתוב פעולה רקורסיבית שתבנה מספר שלם מסדרה של תווי ספרות המתקבלים כקלט. סדרת הספרות לבניית המספר תסתיים כאשר הקלט יהיה תו שאינו ספרה.

10. כתוב פעולה רקורסיבית שהופכת מספר עשרוני למספר אוקטאלי (בסיס 8).

תהליך ההפיכה ממספר עשרוני למספר אוקטאלי:

יש לחלק את המספר העשרוני ל-8. השארית הינה הספרה הימנית של המספר האוקטאלי. מפעילים את הפעולה שוב על החלק המנה. התהליך מסתיים כאשר המנה המתקבלת היא 0 (אפס).

דוגמא:  $4235/8=529(3)$ ,  $529/8=66(1)$ ,  $66/8=8(2)$ ,  $8/8=1(0)$ ,  $1/8=0(1)$   
ומכאן:  $4235_{10} = 10213_8$

11. כתוב פעולה רקורסיבית לביצוע האלגוריתם של אוקלידס. האלגוריתם מיועד למציאת המחלק

המשותף הגדול ביותר (ממ"ג) של שני מספרים טבעיים n ו-m.

ממ"ג = המספר הגדול ביותר המחלק גם את n וגם את m.

12. מספר האפשרויות לבחירת k עצמים מתוך קבוצה של n עצמים שונים מסומנת:  $\binom{n}{k}$   
(n מעל k), ומחושבת כך:  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$  כאשר מוגדר:  $\binom{n}{n} = \binom{n}{0} = 1$ .

כתוב פעולה רקורסיבית לחישוב מספר זה לפי n ו-k.

13. לפניך הליך רקורסיבי בשם `recExample`:

```
public static void recExample (int n1, int n2 )
{
    if (n1 == n2) System.out.println ("stop");
    else
    {
        System.out.println (n1);
        recExample (n1 - 1, n2);
        System.out.println (n1);
    }
}
```

א. מה יהיה הפלט להוראה: `recExample (4,1)`;

ב. נשנה את תנאי העצירה בלבד (שאר הפעולה ללא שינוי). מה יקרה עתה להוראה:

`recExample (4,1)`;

- (1) **if** (n1 > n2) ...
- (2) **if** (n1 < n2) ...
- (3) **if** ((**int**)sqrt(n1) == n2) ...
- (4) **if** (n1 + n2 == 4.5) ...

14. כתוב הליך בשם *bottle* המקבל כפרמטר תו *tav* ומספר אי-זוגי *n* (הנח המספר אי-זוגי). ההליך ידפיס את *tav* על פי התבנית הבאה:

<i>bottle</i> ('@', 3);	<i>bottle</i> ('*', 1);	<i>bottle</i> ('\$ ', 5);
@ @ @	*	\$ \$ \$ \$ \$
@		\$ \$ \$
@ @ @		\$
		\$ \$ \$
		\$ \$ \$ \$ \$

15. לפניך פעולה רקורסיבית:

```
public static int func (int n)
{
    if (n <= 0) return 2;
    return (func (n-1) * func (n-1));
}
```

מה יהיה הפלט כתוצאה ממשפט הזימון: `System.out.println (func(3));`

16. לפניך הליך רקורסיבי:

```
public static void demo (int a, int d, int n)
{
    if (n == 0) System.out.println ("finish");
    else
    {
        System.out.print (a + " ");
        demo (a+d, d, n-1);
    }
}
```

א. מה יהיה הפלט לכל אחת מההוראות הבאות:

- (1) `demo (100, -5, 6);`
- (2) `demo (-6, 3, 4);`
- (3) `demo (2, 1, 5);`

ב. כתוב את ההליך מחדש ללא שימוש ברקורסיה.

17. לפניך פעולה רקורסיבית בשם *wow* :

```
public static void wow (int n)
{
    if (n > 1) wow (n / 2);
    System.out.print (n + " ");
}
```

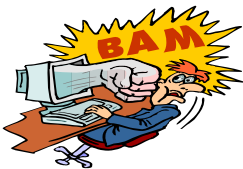
איזו מ-6 התוצאות שלפניך תהווה הפלט למשפט הזימון: *wow* (16);

(1) 10 8 6 4 2	(2) 1 2 4 8 16	(3) 16 8 4 2 4 8 16
(4) 16 8 4 2 1	(5) 2 4 8 16 32	(6) 1 2 4 6 8 10 12 14 16

18. נתונה פעולה רקורסיבית :

```
public static int func (int k, int n)
{
    if (n == k)
        return k;
    if (n > k)
        return func (k, n - k);
    return func (k - n, n);
}
```

א. מה תחזיר הפעולה בתגובה לקריאה: *func* (6,8);  
 ב. הסבר במילים מה מבצעת הפעולה.



הילה **עבודה פורייה**