

## **חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"ה**

ניתן להשתמש בחומרים לצורך הוראה בלבד.

**לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי**

ללא קבלת אישור מראש מצוות הפיתוח

**כתיבה ועריכה:**

**דפנה מינסטר, שרה פונק**



תאריך: \_\_\_\_ . \_\_\_\_ . \_\_\_\_

מקצוע: מדעי המחשב ב' - JAVA

שם: \_\_\_\_\_

נושא: רְקוּרְסִיָּה



## ⌘ דף עבודה מס' 1 – רְקוּרְסִיָּה – פעולות שלא מחזירות ערך ⌘

\*\*\*\*\*

1. לפניך פעולה בשפת JAVA:

```
public static void series1 ( int n )
{ // הפעולה מקבלת מספר טבעי n .
  // הפעולה מחשבת .....
  if ( n >= 10 )
    System.out.print ( n ) ;
  else
  {
    System.out.print ( n + " " ) ;    ← שורת הדפסה
    series1 ( n + 1 ) ;
  }
}
```

לפניך תכנית ראשית בשפת JAVA:

```
public static void main ( String [ ] args )
{
  series1 ( 1 ) ;
}
```

א. נסה / י לעקוב אחר ביצוע התכנית הראשית, רשום/י מהו הפלט שיתקבל לדעתך:

ב. כעת בדק/י בעזרת הרצה במחשב מהו הפלט ? \_\_\_\_\_

ג. הרץ / י את התכנית עבור הקלטים הבאים, ורשום / י את הפלט בטבלה הבאה:

mis2	פלט
<b>series1</b> ( 2 ) ;	
<b>series1</b> ( 7 ) ;	
<b>series1</b> ( 10 ) ;	
<b>series1</b> ( 11 ) ;	

ד. בתכנית שלעיל מופיעה דרך חדשה. נסה/י להסביר במילים את הדרך וכיצד היא מתבצעת.

---

---

---

---



2. לפניך פעולה `series2` בשפת JAVA :  
שים לב! לשינוי ...

```
public static void series2 ( int n )  
{  
    if ( n >= 10 )  
        System.out.print ( n ) ;  
    else  
    {  
        series2 ( n + 1 ) ;  
        System.out.print ( n + " " ) ;  
    }  
}
```

שורת הדפסה ←

א. נסה / י לעקוב אחר ביצוע התכנית הראשית, רשום/י מהו הפלט שיתקבל לדעתך :

ב. כעת בדק/י בעזרת הרצה במחשב מהו הפלט ?

ג. מדוע, לדעתך, הגיב המחשב בצורה זו ?

---

---

ד. הרץ / י את התכנית עבור הקלטים הבאים, ורשום / י את הפלט בטבלה הבאה :

mis2	פלט
<code>series2 ( 2 ) ;</code>	
<code>series2 ( 7 ) ;</code>	
<code>series2 ( 10 ) ;</code>	
<code>series2 ( 11 ) ;</code>	

ה. מה יקרה אם נשנה את התנאי בפעולה שלפניך ל-  $n = 10$  ) ונריץ את התכנית הראשית עם הזימון : `series2 ( 11 ) ;` ?

---

---

---

---

הסבר/י :

---

---

---

---



הגדרה:



\*\*\*\*\*  
\* **רקורסיה** היא דרך חדשה לפתרון בעיות. משימה שקוראת לעצמה.  
\*\*\*\*\*

תנאים לקיום רקורסיה:

1. הליך **רקורסיבי** - תת-משימה המזמנת אותה תת-משימה, עבור ערך קטן/גדול יותר.
2. **תנאי עצירה** – עוצר את פעולת הרקורסיה (היעדרו של תנאי עצירה יוביל לחזרה אינסופית של הרקורסיה).

הרקורסיה מייצרת שני מצבים:

1. שלב **פתיחת** הרקורסיה – הפעולות המתבצעות לפני הזימון הרקורסיבי (עד תנאי העצירה).  
הערכים מעובדים בסדר רגיל.
2. שלב **סגירת** הרקורסיה – הפעולות המתבצעות לאחר הזימון הרקורסיבי.  
הערכים מעובדים בסדר הפוך.

דוגמה לתרגיל:

```
public static void series ( int n )
{
    if ( n >= 7 )
        System.out.print ( n );
    else
    {
        System.out.print ( n + " " );
        series ( n + 1 );
        System.out.print ( n + " " );
    }
}
```

← תנאי עצירה

← שלב פתיחת הרקורסיה  
הערכים מעובדים בסדר רגיל

← זימון/קריאה רקורסיבית

← שלב סגירת הרקורסיה  
הערכים מעובדים בסדר הפוך

⇒ עבור הקלט: n = 1, ייתקבל הפלט: **1 2 3 4 5 6 7 6 5 4 3 2 1**  
} פתיחה } סגירה

3. כתבי פעולה רקורסיבית לכל אחת מהסדרות הבאות:

- א. 1, 4, 7, 10, 13, 16, 19, 22 (קיימים 2 פתרונות שונים)
- ב. -1, -2, -3, -4, -5, -6, -7, -8, -9, -10 (קיימים 2 פתרונות שונים)
- ג. 1, 1, 2, 2, 3, 3, 4, 4, 3, 3, 2, 2, 1, 1
- ד. 1, 2, 3, 4, 3, 2, 1, 10, 20, 30, 40
- ה. א, ב, ג, ד, א, ב, ג, א
- ו. 1, 2, 3, 4, 5, 6, 7, 8, 9, 8, 6, 4, 2





4. נתונות 2 פעולות והזימון שלהן בתכנית הראשית:

<pre>public static void <b>seriesA</b> (int n) {     if ( n &lt;= 1 )         System.out.print ( n + " " );     else     {         System.out.print ( n + " " );         <b>seriesA</b> ( n - 1 );     } }</pre>	<pre>public static void <b>seriesB</b> (int n) {     if ( n &gt;= 4 )         System.out.print ( n + " " );     else     {         <b>seriesB</b> ( n + 1 );         System.out.print ( n + " " );     } }</pre>
<b>seriesA</b> ( 4 ); הזימון	<b>seriesB</b> ( 1 ); הזימון

א. הסברי/י ב- 2-3 שורות מה מבצעת הפעולה **seriesA**:

---



---



---

ב. האם הפעולות הנ"ל יציגו פלט זהה ?

ג. הסברי/י את ההבדל בין הפעולות הנ"ל:

---



---



---

\* 5. חזרי/י לתרגיל 3 סעיפים א' ו- ב' והוסף/י את הפתרון השני לסדרה.

6. פעולה רקורסיבית שמזמנת פעולה רקורסיבית:

א. כתבי/י פעולה רקורסיבית בשם **printStars** המקבלת מספר שלם N ומדפיסה N כוכביות על המסך (כתבי/י פעולה במחלקה).

ב. כתבי/י פעולה רקורסיבית בשם **printRowStars** המקבלת מספר שלם M ומדפיסה N שורות של כוכביות על המסך, כאשר מספר הכוכביות בכל שורה גדל ב- 1.

שימו/י לב ! יש להשתמש בפעולה שכתבת בסעיף א'.

ג. כתבי/י תכנית ראשית הקולטת מספר שלם N ומזמנת את הפעולה **printRowStars**.

דוגמה: עבור הקלט N = 6, ייתקבל הפלט:


```
*
**
***
****
*****
*****
```



פתרון לתרגיל 6 :

<pre>public static void <b>printStars</b> ( int n ) {     if ( n &lt; 1 )         System.out.println ( ) ;     else     {         System.out.print ( ' * ' ) ;         <b>printStars</b> ( n - 1 ) ;     } }</pre>	<pre>public static void <b>printRowStars</b> ( int m ) {     if ( m &lt; 1 )         System.out.print ( m + " " ) ;     else     {         <b>printRowStars</b> ( m - 1 ) ;         <b>printStars</b> ( m ) ;     } }</pre>
--	---

7. כתבי/י פעולה רקורסיבית בשם **multiplicationTable** המדפיסה את לוח הכפל.

רמז: רצוי לחלק את המשימה לתת-משימות. 

8. נתונות 2 פעולות (אחת רקורסיבית) והזימון של הפעולה **printA** בתכנית הראשית:

<pre>public static void <b>print</b> ( int[ ] A , int i ) {     if ( i &gt;= A.length )         System.out.println ( " &lt;-- המערך " ) ;     else     {         System.out.print ( A[ i ] + " " ) ;         <b>print</b> ( A , i + 1 ) ;     } }</pre>	
<pre>public static void <b>printA</b> ( int[ ] A ) {     <b>print</b> ( A , 0 ) ; }</pre>	
<pre>int [ ] a = { 1 , 2 , 3 , 4 , 5 , 6 } ; <b>printA</b> ( a ) ;</pre>	הזימון בתכנית הראשית:

א. מה יהיה הפלט של הזימון **print** ( A , 0 ) ?

---

---

ב. הסברי/י ב- 2-3 שורות מה מבצעת הפעולה **print** :

---

---

---



9. כתבי/ פתרון רקורסיבי נוסף לפעולה **print** (מתרגיל 8).  
10. נתונות 2 פעולות והזימון שלהן בתכנית הראשית:

```
public static void printString1 ( String S , int L , int H )
{
    if ( L > H )
        System.out.print ( " " );
    else
        if ( L == H )
            System.out.print ( S.charAt( L ) + " " );
        else
            {
                System.out.print ( S.charAt( L ) + " " + S.charAt( H ) + " " );
                printString1 ( S , L + 1 , H - 1 );
            }
}
```

שימי לב!

אסור לזמן את הפעולה הרקורסיבית עם  $L++$  או  $H--$ , כי זו רקורסיה אינסופית.

**String s = "123456" ;**

**הזימון בתכנית הראשית:**

**printString1 ( s , 0 , s.length() - 1 ) ;**

```
public static void printString2 ( String S )
{
    if ( S.length() == 0 )
        System.out.print ( " " );
    else
        if ( S.length() == 1 )
            System.out.print ( S.charAt( 0 ) + " " );
        else
            {
                System.out.print ( S.charAt( 0 ) + " " + S.charAt( S.length() - 1 ) + " " );
                printString2 ( S.substring( 1 , S.length() - 1 ) );
            }
}
```

**String s = "123456" ;**

**הזימון בתכנית הראשית:**

**printString2 ( s ) ;**

א. מה יהיה הפלט של הזימון `printString1( s , 0 , s.length() - 1 ) ;` ?

ב. הסברי/י ב- 2-3 שורות מה מבצעת כל אחת מהפעולות הנ"ל:

ג. בחרי/י את הדרך המועדפת עליך, נמקי/י תשובתך



11. לפניך פעולה בשפת JAVA:

```
public static void printM ( int [ ] [ ] mat , int i , int j )
{
    if ( i >= mat.length )
        System.out.println ( ) ;
    else
        if ( j >= mat[ i ].length )
        {
            System.out.println ( ) ;
            printM ( mat , i + 1 , 0 ) ;
        }
        else
        {
            System.out.print ( mat[ i ][ j ] + " " ) ;
            printM ( mat , i , j + 1 ) ;
        }
}
```

10	20	30
40	50	60
70	80	90
100	110	120

mat

א. מה יהיה הפלט של הזימון ? printM ( mat , 0 , 0 ) ;

---

---

---

---

---

ב. הסברי/י ב- 2-3 שורות מה מבצעת הפעולה printM :

---

---

---



עבודה נעימה !

