# תבניות אלגוריתמיות לפעולות רקורסיביות

## תבנית צבירה

תבנית צבירה לסדרת ערכים:

**סכום** (סדרת ערכים)

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   החזר את ערך הקצה + **סכום** (סדרת הערכים ללא ערך הקצה)

דוגמא:

/* פעולה המחזירה את סכום הספרות שבמספר שלם לא שלילי num */
**סכום-ספרות** (num)

1. אם num < 10 החזר num או אם num = = 0 החזר 0
2. החזר את ספרת האחדות של num + **סכום-ספרות** (num ללא ספרת האחדות)

```
public static int sumOfDigits (int num)
{
	if (num < 10) return num;		//---  if (num = = 0) return 0;
	return  num % 10 + sumOfDigits (num / 10);
}
```

## תבנית מניה

תבנית מניה למספר הערכים בסדרת ערכים

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   החזר 1 (בגלל ערך הקצה) + **סכום** (סדרת הערכים ללא ערך הקצה)

דוגמא:

/* פעולה המחזירה את מספר הספרות שבמספר שלם לא שלילי num */
**מספר-ספרות** (num)

1. אם num < 10 החזר 1 או אם num = = 0 החזר 0
2. החזר 1 + **מספר-ספרות** (num ללא ספרת האחדות)

```
public static int numOfDigits (int num)
{
	if (num < 10) return 1;		//---  if (num = = 0) return 0;
	return  1 + numOfDigits (num / 10);
}
```

## I צבירה אותנית

דרך א׳:

**סכום-מותנה** (סדרת ערכים)

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   2.1 אם ערך הקצה מקיים את תנאי החיפוש
   החזר: ערך הקצה + **סכום-מותנה** (הערכים ללא ערך הקצה)
   2.2 אחרת - החזר: **סכום-מותנה** (סדרת הערכים ללא ערך הקצה)

/* פעולה המחזירה את סכום הספרות הגדולות מ- 5 שבמספר שלם לא שלילי num */

**סכום-ספרות-גדול-מ-5** (num)

1. אם num = = 0 החזר 0
2. אם num % 10 > 5
   החזר: num % 10 + **סכום-ספרות-גדול-מ-5** (num/10)
3. החזר: **סכום-ספרות-גדול-מ-5** (num/10)

דרך ב׳: (שימוש במשתנה עזר)

**סכום-מותנה** (סדרת ערכים)

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   2.1 אם ערך הקצה מקיים את תנאי החיפוש
   ערך הקצה ← z
   2.2 אחרת
   z ← 0
   2.3 החזר: z + **סכום-מותנה** (הערכים ללא ערך הקצה)

/* פעולה המחזירה את סכום הספרות הגדולות מ- 5 שבמספר שלם לא שלילי num */

**סכום-ספרות-גדול-מ-5** (num)

1. אם num = = 0 החזר 0
2. אם num % 10 > 5 z ← num % 10
   אחרת - z ← 0
3. החזר: z + **סכום-ספרות-גדול-מ-5** (num/10)

## II מניה אותנית

דרך א':

**מניה-מותנית** (סדרת ערכים)

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   2.1 אם ערך הקצה מקיים את תנאי החיפוש
   החזר: 1 + **מניה-מותנית** (הערכים ללא ערך הקצה)
   2.2 אחרת - החזר: **מניה-מותנית** (סדרת הערכים ללא ערך הקצה)

/* פעולה המחזירה את מספר הספרות הזוגיות שבמספר שלם לא שלילי num */

**מספר-ספרות-זוגיות** (num)

1. אם num = = 0 החזר 0
2. אם num % 2 = = 0
   החזר: 1 + **מספר-ספרות-זוגיות** (num/10)
3. החזר: **מספר-ספרות-זוגיות** (num/10)

דרך ב':

**מניה-מותנית** (סדרת ערכים)

1. אם מתקיים תנאי הסיום
   החזר ערך
2. אחרת -
   2.1 אם ערך הקצה מקיים את תנאי החיפוש
   $z \leftarrow 1$
   2.2 אחרת
   $z \leftarrow 0$
   2.3 החזר: z + **מניה-מותנית** (סדרת הערכים ללא ערך הקצה)

/* פעולה המחזירה את מספר הספרות הזוגיות שבמספר שלם לא שלילי num */

**מספר-ספרות-זוגיות** (num)

1. אם num = = 0 החזר 0
2. אם num % 2 = = 0 $z \leftarrow 1$
   אחרת $z \leftarrow 0$
3. החזר: z + **מספר-ספרות-זוגיות** (num/10)

## III לביה במערך

סכום-איברי-המערך (arr, place)

1. אם מתקיים תנאי הקצה – החזר את איבר הקצה
2. החזר את האיבר שבקצה + סכום-איברי-המערך (המערך ללא האיבר שבקצה)

```
/*        פעולה המחזירה את סכום איברי המערך        */
```

| | |
|---|---|
| ***public*** **static int** sumOfArr (**int** [] arr)<br>{<br>**return** sumOfArr (arr, 0) ;<br>} | ***public*** **static int** sumOfArr (**int** [] arr)<br>{<br>**return** sumOfArr (arr, arr.length - 1) ;<br>} |
| ***private*** **static int** sumOfArr (**int** [] arr, **int** place)<br>{<br>**if** (place = = arr.length)<br>**return** 0 ;<br>**return** arr[place] +<br>sumOfArr (arr, place + 1);<br>} | ***private*** **static int** sumOfArr (**int** [] arr, **int** place)<br>{<br>**if** (place = = 0)<br>**return** arr[place] ;<br>**return** arr[place] +<br>sumOfArr (arr, place - 1);<br>} |

**שימו ♥ :**

משפט הזימון מבקש לבדוק את סכום איברי המערך.

לצורך המימוש אנו זקוקים למקומו של האיבר שבקצה המערך (האיבר שנמצא במקום הראשון או האחרון), דרך המימוש מוסתרת מהמשתמש בפעולה.

ולכן הפעולה **סכום-המערך (arr)** הינה פעולה גלויה – public
ואילו הפעולה **סכום-המערך** (arr, place) היא פעולה מוסתרת – privae

הגדרת שתי הפעולות, במחלקת השירות המטפלת במערך, אפשרית מכיוון שלכל אחת מהן חתימה שונה (חתימה – כותרת הפעולה, כולל הפרמטרים).

<u>מושגים</u> :

הסתרת מידע – Encapsulation

העמסת פעולות - Methods Overloading

## IV מניה אותנית מערך

**מספר-האיברים** (arr, place)

1. אם מתקיים תנאי הקצה – החזר את איבר הקצה
2. אם האיבר שבקצה מקיים את תנאי הבדיקה,
   החזר 1 + **מספר-האיברים** (המערך ללא איבר הקצה)
   החזר **מספר-האיברים** (המערך ללא איבר הקצה)

/* פעולה המחזירה את מספר איברי המערך האי-זוגיים */

```
public static int numOfOddNumbers (int [] arr)
{
        return numOfOddNumbers (arr, 0) ;
}
```

```
public static int numOfOddNumbers (int [] arr)
{
        return
          numOfOddNumbers (arr, arr.length - 1) ;
}
```

```
private static int numOfOddNumbers
          (int [] arr, int place)
{
    if (place = = arr.length) return 0 ;
    if (arr[place] % 2 = = 1)
       return  1 +
             numOfOddNumbers(arr, place+1);
    return
       numOfOddNumbers(arr, place+1);

}
```

```
private static int numOfOddNumbers
          (int [] arr, int place)
{
    if (place < 0) return 0 ;
    if (arr[place] % 2 = = 1)
       return  1 +
             numOfOddNumbers(arr, place-1);
    return
       numOfOddNumbers(arr, place-1);

}
```

<u>הערה</u>:
במקרה של מניה (או צבירה) מותנית, נעדיף לקבוע את תנאי הקצה אחרי האיבר הראשון (או האחרון) כדי לחסוך בבדיקה של איבר זה.
דוגמה: אם תנאי הקצה היה if (place == 0)
היה עלינו לבדוק את הזוגיות של האיבר במקום 0 כדי לקבוע איזה ערך יוחזר