

## שאלות חזרה

1. כתבו פונקציה שמקבלת מערך (arr) של מספרים שלמים, הפונקציה מחזירה אחוז של מספרים ראשוניים הנמצאים במערך.

```
public static double percentPrimes(int[] arr) {...}
                                     דוגמה:
int arr[] = {22,30,5,8,16,7,10,9,13,17};
                                     התוצאה: ans:40%
```

2. כתבו פונקציה שמקבלת מספר שלם חיובי (int) ומחזירה סכום הספרות הזוגיות שלו.

```
public static int sumEvenDig(int num) (int n) {...}
                                     דוגמה: num = 142458, התוצאה: 18
```

3. כתוב פונקציה שמקבלת מחרוזת של אותיות לטיניות קטנות ומחזירה אות המופיע הכי הרבה פעמים:

```
public static char freqLetter (String s)
```

4. כתבו פונקציה שמקבלת מערך של מספרים ממשיים ומספר ממשי כלשהו. הפונקציה מחזירה מערך של איברי המערך הנתון הגדולים או שווים a.

```
public static double[] biggerA(double[] arr, double a){...}
                                     דוגמה: קלט: double arr[]={1, 2.5, 3, 4.5, 5.2} , a=4.1
                                     פלט: {4.5, 5.2}
```

5. כתבו פונקציה שמקבלת מספר שלם  $n$  ( $1 \leq n \leq 18$ ), ומחזירה את המספר הדו-ספרתי הגדול ביותר שסכום ספרותיו שווה למספר זה. הנחה: קלט תקין ( $1 \leq n \leq 18$ )

```
int maxSumDigits (int n) {...}
```

דוגמה: קלט: 16, פלט: 97

6. כתבו פונקציה שמקבלת מחרוזת ומחזירה כמה תווים שונים יש בה.

```
int diffLetters(String s) {...}
```

7. כתבו פונקציה סטטית שמקבלת אוסף PointContainer ומחזירה אוסף חדש (העתקה עמוקה) של נקודות שמכיל את כל הנקודות באוסף המקורי שנמצאות ברביע הראשון. שימו לב שני האוספים חייבים להיות בלתי תלויים בזיכרון

```
PointContainer q1(PointContainer pc) {...}
```

8. הוסף למחלקה PointContainer שיטה שמקבלת אוסף נוסף ומחזירה אוסף חדש שמכיל את חיתוך האוספים.

```
PointContainer intersection(PointContainer pc) {...}
```

9. התייחסו לקוד המיון שמופיע בדוגמאות הקוד למבחן, בצעו את השינויים הרלוונטיים כך שהפונקציה sort תמיין מערך של נקודות (לפי מרחקן מראשית הצירים).

10. כתוב מחלקה Circle (מעגל) שתכונותיה נקודת מרכז של מעגל (מסוג נקודה) ורדיוסו שהוא מספר ממשי:

```
public class Circle {  
    private Point _center;  
    private double _radius;  
    . . .  
}
```

הוסף שיטות הבאות:

- בנאי,
- בנאי מעתיק,
- שיטה שמקבלת מעגל ושמחזירה 0 אם המעגלים אינם נחתכים, 1 אם יש רק נקודת השקה ומחזירה 2 אם יש שתי נקודות חיתוך:

```
public int intersection(Circle c)
```

הנחיה:

אם סכום של הרדיוסים קטן ממרחק בין מרכזי המעגלים – המעגלים אינם נחתכים.  
אם סכום של הרדיוסים גדול ממרחק בין מרכזי המעגלים – המעגלים נחתכים, ויש שתי נקודות חיתוך.  
אם סכום של רדיוסים שווה למרחק בין נקודות מרכז – למעגלים יש נקודת השקה.

