

## **קרא בעיון את ההנחיות שלהלן:**

- כל התכניות צריכות להיות מתועדות היטב. יש לכתוב תחילה **בקצרה** את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית. יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתכנית. תכנית שלא תתועד כנדרש לעיל תקבל לכל היותר 85% מהניקוד.  
**אפשר לתעד בעברית.** אין צורך בתיעוד API.
- יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה. **תכנית לא יעילה לא תקבל את מלוא הנקודות.**
- אם ברצונך להשתמש בתשובתך בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיק את השיטה או את המחלקה למחברת הבחינה. מספיק להפנות למקום הנכון, ובלבד שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').
- אין להשתמש במחלקות קיימות ב-Java, חוץ מאלו המפורטות בשאלות הבחינה.
- יש לשמור על סדר; תכנית הכתובה בצורה בלתי מסודרת עלולה לגרוע מהציון.
- בכתובת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה. אין להשתמש במשתנים גלובליים!

<ul style="list-style-type: none"><li>• <b>את התשובות לשאלות 3 - 7 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!</b></li></ul>
---

**חומר עזר המותר בשימוש הוא:**

**1. חוברת השקפים של הקורס**

**2. ספר הלימוד Java Software Solutions**

**אסור להשתמש במחשב מכל סוג שהוא!**

**חלק א – עליכם לענות על כל השאלות בחלק זה במחברת הבחינה**

## שאלה 1 - 25 נקודות

נתונה מחרוזת תוים  $s$ . במהלך שליחתה ברשת האינטרנט קרתה תקלה וחלק מהתווים במחרוזת הוכפלו והופיעו ברצף מספר לא ידוע של פעמים. כלומר במחרוזת  $t$  שהתקבלה הופיעו כל התווים של  $s$ , לפי הסדר במחרוזת המקורית, אבל לפעמים תו מסוים לא הופיע פעם אחת כמו במחרוזת  $s$  אלא ברצף מספר לא ידוע של פעמים.

**נגדיר** שמחרוזת תוים  $t$  **עברה-טרנספורמציה** ממחרוזת התווים  $s$ , אם כל תו של  $s$  נמצא ב- $t$  (לפי הסדר ב- $s$ ) **לפחות** פעם אחת.

שימו לב שאם תו מופיע במחרוזת  $s$  כמה פעמים  $(n_k)$ , אז במחרוזת  $t$  שעברה טרנספורמציה הוא יופיע לפחות  $k$  פעמים.

**לדוגמא:**

s = "abbc**d**" אם המחרוזת

אז כל המחרוזות הבאות **עברו-טרנספורמציה** ממחרוזת זו :

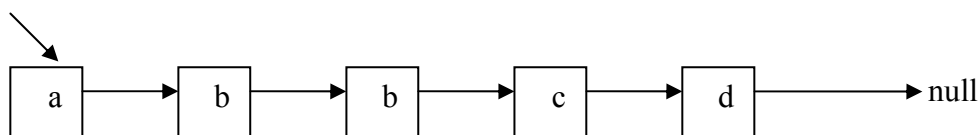
"abacd" , "aaaabacd" , "abacdcccccc" , "aabaccdd" , "abbbacd"

וכל המחרוזות הבאות לא עברו-טרנספורמציה מהמחרוזת s:

"a" , "abcd" , "aaccbbdd"

אנו נייצג מחרוזת תווים באמצעות רשימה מקושרת חד-סטרית, כך שכל צומת ברשימה מכיל תו אחד מהמחרוזת.

כך, למשל, מחרוזת התווים  $s$  שלעיל תיוצג כך על-ידי הרשימה המקושרת:



להלן המחלקה CharNode המייצגת תו בודד.

```
public class CharNode {
    private char _value;
    private CharNode _next;
    public CharNode(char val, CharNode n) {
        _value = val;
        _next = n;
    }
}
```

```

    public char getValue() {
        return _value;
    }
    public CharNode getNext() {
        return _next;
    }
    public void setValue(char v) {
        _value = v;
    }
    public void setNext(CharNode node) {
        _next = node;
    }
}

```

המחלקה CharList הנתונה להלן מייצגת מחרוזת תווים על ידי שימוש ברשימה מקושרת:

```

public class CharList {
    private CharNode _head;

    public CharList( ) {
        _head = null;
    }
}

```

אתם יכולים להניח שהרשימה כבר מייצגת מחרוזת תווים ואין צורך לבצע זאת.

הוסיפו למחלקה CharList שיטה **סטטית** בוליאנית **רקורסיבית** שחתימתה היא:

```
public static boolean isTrans (CharList list1, CharList list2)
```

המקבלת כפרמטרים שתי רשימות list1 ו-list2, המייצגות בהתאמה את מחרוזות התווים s ו-t.

השיטה צריכה להחזיר true אם t עברה-טרנספורמציה מהמחרוזת s, ו-false אחרת.

**השיטה צריכה להיות רקורסיבית ללא שימוש בלולאות כלל. כך גם כל שיטות העזר שתכתבו (אם תכתבו) לא יכולות להכיל לולאות.**

**אפשר להשתמש בהעמסת-יתר (overloading).**

**שימו לב, השיטה חייבת לעבוד על הרשימה. אי אפשר להעביר את הרשימות למחרוזות, למשל, לבדוק אותן ואז להחזיר תשובה.**

**אל תשכחו לתעד את מה שכתבתם!**

## שאלה 2 - 25 נקודות

נתונה המחלקה IntVector הבאה המממשת מערך חד ממדי המכיל מספרים שלמים.

```
public class IntVector
{
    private int []_arr;

    /**
     * Constructor for objects of class IntVector
     */
    public IntVector(int size)
    {
        _arr= new int[size];
    }

    ...    //methods
}
```

אתם יכולים להניח שהמערך מלא במספרים (חיוביים ושליליים).

כמו כן, נתונה במחלקה IntVector השיטה הבאה:

```
public int what ()
{
    int m =_arr[0];
    for (int i=0; i<_arr.length; i++)
        for (int j=i; j<_arr.length; j++)
        {
            int s=_arr[i];
            for (int k=i+1; k<=j; k++)
                s += _arr[k];
            if (s > m)
                m=s;
        }
    return m;
}
```

א. מה מבצעת השיטה what? הסבירו בקצרה מה מבצעת השיטה, ולא כיצד היא מבצעת זאת. (6 נק')

ב. מה סיבוכיות זמן הריצה של השיטה? הסבירו תשובתכם. (3 נק')

ג. כתבו את השיטה what כך שתבצע את מה שביצעה בסעיף א' בסיבוכיות זמן ריצה קטנה יותר. (13 נק')

השיטה שתכתבו צריכה להיות יעילה ככל הניתן. תשובה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה לא תקבל את מלוא הנקודות.

ד. מה סיבוכיות זמן הריצה של השיטה שכתבתם בסעיף ג? הסבירו תשובתכם. (3 נק')

אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3-7 יש לכתוב על גבי השאלון.  
לא נבדוק תשובות שייכתבו במקום אחר!**

**שאלה 3 - 10 נקודות**

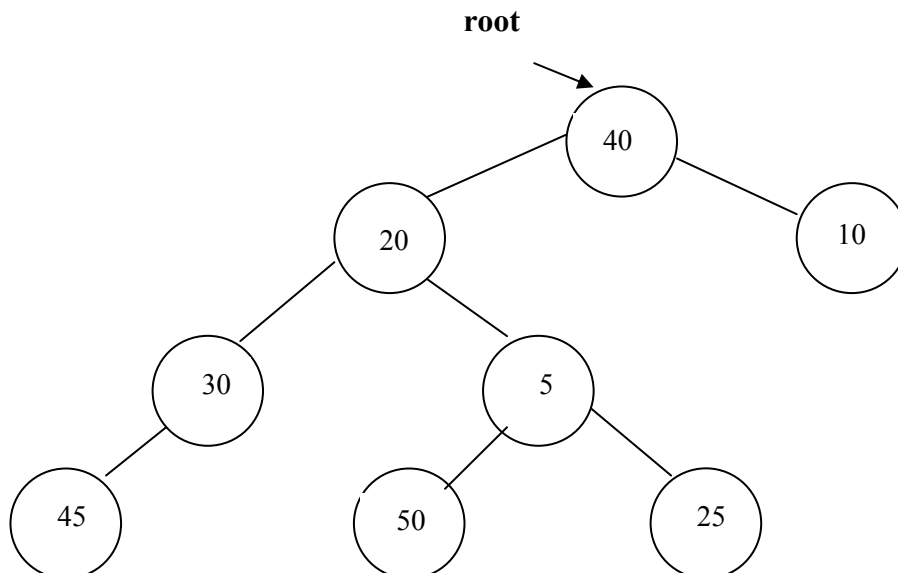
נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int number) {
        _number = number;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }
}
```

נתון העץ הבינרי הבא, ששורשו הוא root



להלן נתונות שיטות סטטיות שונות, המבצעות פעולות שונות על עצים בינריים.  
השיטה min מקבלת כפרמטרים שני מספרים שלמים a ו-b ומחזירה את הקטן ביניהם. הניחו שהמימוש שלה כבר כתוב.  
השיטה isLeaf מקבלת כפרמטר צומת בעץ בינרי ומחזירה true אם הוא עלה (כלומר, צומת ללא בנים) ו-false אחרת.

```
public static int min (int a, int b) { ... }
public static boolean isLeaf (Node t) { ... }

public static int f (Node t)
{
    if (t == null)
        return 0;
    return 1 + min (f (t.getLeftSon()), f (t.getRightSon()));
}

public static Node what (Node t)
{
    return what (t, f(t));
}

private static Node what (Node t, int num)
{
    if ((t == null) || ((num == 1) && (isLeaf(t))))
        return t;
    Node temp = what (t.getLeftSon(), num-1);
    if (temp != null)
        return temp;
    return what (t.getRightSon(), num-1);
}
```

### סעיף א. (3 נקודות)

מה מבצעת השיטה f בהינתן לה כפרמטר t מטיפוס Node שמצביע לשורש של עץ בינרי? כתבו בקצרה מה מבצעת השיטה ולא **כיצד** היא מבצעת זאת.

---



---



---

### סעיף ב. (2 נקודות)

איזה צומת תחזיר השיטה **what** בהינתן לה כפרמטר root מטיפוס Node שמצביע לשורש של העץ בינרי המצויר לעל?

---

### סעיף ג. (5 נקודות)

מה מבצעת השיטה **what** בהינתן לה כפרמטר t מטיפוס Node שמצביע לשורש של עץ בינרי? כתבו בקצרה מה מבצעת השיטה ולא **כיצד** היא מבצעת זאת.

---

---

---

### שאלה 4 - 5 נקודות

לפניכם קטע הקוד הבא:

```
for (int i=1; i<n; i++)
{
    for (int j=i; j<=m; j++)
        S1;
    for (int k=t; k>=i; k--)
        S2;
}
```

כאשר S1 ו-S2 הם משפטים פשוטים והזמן הדרוש לביצועם הוא  $O(1)$ . כמו כן, S1 ו-S2 אינם משנים את הערכים של  $i, j, k, n, m, t$ . מה סיבוכיות זמן הריצה של קטע הקוד הנתון? הקיפו בעיגול את התשובה הנכונה.

1.  $O(\max(n, m, t))$

2.  $O(n*m*t)$

3.  $O(n+m*t)$

4.  $O(n*(m+t))$

## שאלה 5 - 11 נקודות

נתונה המחלקה Question4 הבאה, המכילה שיטות סטטיות שונות הפועלות על מחרוזות.

```
public class Question4
{
    public static String f1(String s)
    {
        return s.substring(0, s.length()/2);
    }

    public static String f2(String s)
    {
        return s.substring(s.length()/2);
    }

    public static String what (String s1, String s2)
    {
        if (s1.length() == 0)
            return s2;
        if (s2.length() == 0)
            return s1;
        if (s1.charAt(0) < s2.charAt(0))
            return s1.charAt(0) + what (s1.substring(1), s2);
        return s2.charAt(0) + what (s1, s2.substring(1));
    }

    public static String something(String s)
    {
        if (s.length() > 1)
        {
            String s1 = f1(s);
            String s2 = f2(s);
            s1 = something(s1);
            s2 = something(s2);
            return what(s1, s2);
        }
        else
            return s;
    }
}
```



## תזכורת -

- `public char charAt(int i)` - מחזירה את התו במקום ה-`i` במחרוזת עליה היא מופעלת.
- `public int length()` - מחזירה את אורך המחרוזת עליה היא מופעלת.
- `public String substring(int i)` - מחזירה את התת-מחרוזת המתחילה במקום ה-`i` במחרוזת עליה היא מופעלת עד לסוף המחרוזת.
- `public String substring(int i, int j)` - מחזירה את התת-מחרוזת המתחילה במקום ה-`i` במחרוזת עליה היא מופעלת עד לתו במקום ה-`j` (לא כולל `j` עצמו).

## סעיף א (2 נקודות)

מה מבצעות השיטות `f1` ו-`f2` כאשר הן מקבלות כפרמטרים מחרוזת `s`?  
כתבו בקצרה מה מבצעות השיטות ולא **כיצד** הן מבצעות זאת.

\_\_\_\_\_ - `f1`

\_\_\_\_\_ - `f2`

## סעיף ב (4 נקודות)

מה תחזיר השיטה `something` אם היא תקבל כפרמטר את מחרוזת התוים `"apple"`

\_\_\_\_\_

## סעיף ג (5 נקודות)

מה מבצעת השיטה `something` כאשר היא מקבלת כפרמטר מחרוזת תוים `s`?  
כתבו בקצרה מה מבצעת השיטה ולא **כיצד** היא מבצעת זאת.

\_\_\_\_\_

\_\_\_\_\_

## שאלה 6 - 16 נקודות

לפניכם חמש המחלקות הבאות (כל מחלקה בקובץ נפרד, כמובן. מספרנו את השורות לנוחותכם):

```
1. public abstract class Vehicle
2. {
3.     public void sketch()
4.     {
5.         System.out.println ("Vehicle.sketch()");
6.     }
7.
8.     public abstract void move();
9. }
-----
10. public class Truck extends Vehicle
11. {
12.     public void sketch()
13.     {
14.         System.out.println ("Truck.sketch()");
15.     }
16.
17.     public void move()
18.     {
19.         System.out.println ("Truck.move()");
20.     }
21. }
-----
22. public class Car extends Vehicle
23. {
24.     public void move()
25.     {
26.         System.out.println ("Car.move()");
27.     }
28. }
-----
29. public class Sedan extends Car
30. {
31.     public void sketch()
32.     {
33.         System.out.println ("Sedan.sketch()");
34.     }
35.
36.     public void move()
37.     {
38.         System.out.println ("Sedan.move()");
39.     }
40. }
-----
41. public class Bus extends Vehicle
42. {
43.     public void sketch()
44.     {
45.         System.out.println ("Bus.sketch()");
46.     }
47. }
```

בנוסף, נתונה המחלקה המשתמשת בחמש מחלקות אלו:

```

45. public class Driver
46. {
47.     public static void main (String [] args)
48.     {
49.         Vehicle [] s = new Vehicle[5];
50.         s[0] = new Truck();
51.         s[1] = new Vehicle();
52.         s[2] = new Sedan();
53.         s[3] = new Car();
54.         s[4] = new Bus();

55.         for (int i = 0; i < s.length; i++)
56.             s[i].sketch();
57.         System.out.println();
58.         for (int i = 0; i < s.length; i++)
59.             s[i].move();
60.     }
61. }

```

#### סעיף א: (6 נקודות)

בקוד שלעיל (במחלקות השונות) נמצאו שגיאות קומפילציה. כתבו מהן, את המקום המדויק שבו נמצאו וכיצד אפשר לתקן את השגיאות. התיקון צריך להיות כזה שאפשר יהיה להמשיך ולהשתמש במחלקה בה נמצאה השגיאה. מספר השורות בטבלה אינו בהכרח מספר השגיאות.

מיקום	מהות השגיאה	התיקון האפשרי

### סעיף ב: (10 נקודות)

בהנחה שהשגיאות שנמצאו בסעיף א יתוקנו, מה יודפס לאחר הרצת השיטה main במחלקה Driver.

---

---

---

---

---

---

---

---

---

---

---

---

### שאלה 7 - 8 נקודות

במחלקה A מוגדרת שיטה בעלת הכותרת הבאה:

```
public void f (int x)
```

לכל אחד מהסעיפים הבאים כתבו אם השורה הכתובה בו יכולה להיות חתימה חוקית לשיטה במחלקה B היורשת מ-A. אם כן, כתבו אם זו העמסה או דריסה או שיטה אחרת, אם לא, הסבירו מדוע.

\_\_\_\_\_ public void f(int s)      א)

\_\_\_\_\_

\_\_\_\_\_ public int f(int s)      ב)

\_\_\_\_\_

\_\_\_\_\_ public void f(int s, int t)      ג)

\_\_\_\_\_

\_\_\_\_\_ public void f(A x)      ד)

## בהצלחה