

**מזהה קורס: 20441 שם קורס: מבוא למדעי המחשב ושפת Java**

מספר שאלת	נכונות מירבי	ציון	ציוון
1.1	15.00	15.00	15.00
1.2	10.00	10.00	10.00
2	25.00	25.00	25.00
3	20.00	20.00	20.00
4	12.00	14.00	12.00
5	16.00	16.00	16.00

**ציון בבחינה סופי : 98.00****הבחינה הבודקה בעמודים הבאים**

473 מ"ס שאלון -

21

כ' באדר א' תשפ"ב

89 מ"ס מועד

בפברואר 2022

סמסטר 2022 א

20441 / 4

### שאלון בחינת גמר

## Java - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 13 עמודים

### מבנה הבחינה:

- קראו בעיון את ההנחיות שלhallon:
  - \* בבחינה יש חמישה שאלות.
  - \* כל התכניות צריכות להיות מוגדרות היטב.
  - יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
  - יש לבחור בשמות משתנים, לפונקציות ולקבועים שבתכנית.
  - תכנית שלא כתועד כנדרש לעיל תקבל לכל היתר % 85 מהניקוד.
  - \* יש להזכיר לכתוב את התכניות בצורה מבנית ויעילה.
  - תכנית לא יעילה לא תקבל את מלאה הנקודות.
  - \* אם ברצונכם להשתמש בתשובתכם בשיטה או בחלוקת הכתיבה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את המחלוקת למחברת הבחינה. מספיק להפנות למקום הנכון, וב惟ך שההפניה תהיה מדויקת (פרמטרים, מיקום וכו').
  - \* אין להשתמש בחלוקת קיימות ב- Java, חוץ מألو המפורטים בשאלות הבחינה.
  - \* יש לשמור על סדר; תכנית הכתיבה בצורה בלתי מסודרת עלולה לגרוע מהמצוין.
  - \* בכתב התכניות יש להשתמש אר ורך במרכיבי השפה שנלמדו בקורס זה  
אין להשתמש במשתנים גלובליים!
  - \* אפשר לטעוד בעברית. אין צורך בתיעוד API.
- כל התשובות צריכות להיות קובץ המבחן במקומות המתאים בלבד.  
תשובה שתיכתב שלא במקום לא תיבדק.

### חומר עזר:

חווברות השקפים 6-12, 7-12.

אין להכניס חומר מודפס נוספת או חומר אחר מכל סוג.

אין להכניס מחשב או מחשבון או מכשיר אלקטרוני מכל סוג שהוא.

בהצלחה !!!

### הצדiron

למשגיח את השאלון  
וכל עזר אחר שקיבלתם בהתאם לתשובות





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

### שאלה 1 (25 נקודות)

בහינתן שתי מחרוזות תווים str1 ו- str2. נאמר שהמחרוזת str2 מופיעה **בדילוגים קבועים** בתוך המחרוזת str1, אם כל התווים שנמצאים במחרוזת str2 מופיעים בתוך המחרוזת str1, באותו הסדר בו הם מופיעים ב-str2, וגם, ביןתו של str2 בתוך str1 יש אותו מספר תווים שלא קשורים למחרוזת str2.

לדוגמא,

1. המחרוזת "abc" = str2 מופיעה בתוך המחרוזת "adbrfa" = str1 **בדילוגים קבועים** של 2. האותיות של str2 מודגשות.
2. המחרוזת "abcd" = str2 מופיעה בתוך המחרוזת "abcdefa" = str1 **בדילוגים קבועים** של 1. האותיות של str2 מודגשות.
3. המחרוזת "abc" = str2 מופיעה בתוך המחרוזת "adbfbbaagcxabcd" = str1 **פעמיים** בדילוגים קבועים. פעם אחת בדילוגים של 4 (החל מהתא הראשון במחרוזת str1), ופעם שנייה בדילוגים של 1 (מיד אחרי התא 'a'). האותיות של str2 מודגשות.
4. המחרוזת "abc" = str2 **לא** מופיעה בתוך המחרוזת "cbdaadbrsdcfa" = str1 **בדילוגים קבועים**.

שים לב, ברור ש- str1 **צריכה להיות אורך או שווה ל-** str2. **צריך לבדוק זאת!**

בשאלה זו נמצא את הדילוג הקבוע בו נמצאת המחרוזת str2 בתוך str1 (אם יש כזה). **נתיחס רק למחרוזות שמתחלות באותו تو.**

### סעיף א: (15 נקודות)

כתבו שיטה סטטית **rkorsibit** בוליאנית בשם **isJump** המתקבלת כפרמטרים שתי מחרוזות תווים str1, str2 ומספר שלם חיובי step, ומחזירה true אם המחרוזת str2 מופיעה בדילוגים קבועים של step בתוך המחרוזת str1 **החל מהתא הראשון בשתי המחרוזות**, ו- false אחרת.

לדוגמא:

1. "abc" = str2 אם step = 2 str1 = "adbrfa", str2 = "abc" true .
2. "abc" = str2 אם step = 2 str1 = "cbdadbrfa", str2 = "abc" false .  
בדילוג קבוע של 2 בתוך str1, אבל הדילוג לא מתחיל בתוא הראשון של str1.
3. "abc" = str2 אם step = 4 str1 = "adcfbaagcxabcd", str2 = "abc" true , אבל אם step = 1 יוחזר false כי הדילוג הזה לא מתחיל בתוא הראשון של str1.
4. "abc" = str2 אם step = 1 str1 = "abcfbaagcxabcd", str2 = "abc" true , וגם אם step = 4 יוחזר false כי שני הדילוגים מתחלים בתוא הראשון של str1.

התימת השיטה היא:

```
public static boolean isJump (String str1, String str2, int step)
```



## סעיף ב: (10 נקודות)

כתבו שיטה סטטית רקורסיבית `strStep` המתקבל כפרמטרים שתי מחרוזות תווים `str1` ו-`str2` ומחזירה את הערך של הדילוג הקבוע בו נמצאת המחרוזת `str2` בתוך `str1` החל מהຫוו הראשון של `str1`. אם המחרוזת `str2` לא נמצאת בדילוג קבוע בתוך `str1`, או שהדילוג לא מתחילה מהຫוו הראשון של `str1`, השיטה תחזיר `-1`. אם המחרוזת `str2` נמצאת בכמה דילוגים קבועים החל מהຫוו הראשון, השיטה תחזיר את ערך הדילוג הראשון בו `str2` נמצאת ב-`str1`.

לדוגמא, בדוגמאות של סעיף א –

1. השיטה `strStep(str1, str2 = "abc")` תחזיר את הערך `2`.
2. השיטה `strStep(str1 = "cbdadabrcfa", str2 = "abc")` תחזיר את הערך `-1`.
3. השיטה `strStep(str1 = "adcfbaagcxabcd", str2 = "abc")` תחזיר את הערך `4`.
4. השיטה `strStep(str1 = "adcfbaagcxabcd", str2 = "abc")` תחזיר את הערך `1`. (שים לב, יש כאן שני דילוגים, אבל `1` הוא הראשון ולכון שהוא שיוויוני)

חתימת השיטה היא:

```
public static int strStep(String str1, String str2)
```

שםו לב, מותר לכם להשתמש בשיטה `jump` מסעיף א, גם אם לא פתרתם אותה.

השיטות מהמחלקה `String` בהן מותר לכם להשתמש הן:

- `s.length()` – המחזירה את אורך המחרוזת `s`
- `s.charAt(int index)` – המחזירה את התו הנמצא במקומות `index` במחרוזת `s`
- `s.indexOf(char c)` – המחזירה את המיקום הראשון בו נמצא התו `c` במחרוזת `s`
- `(i .. s.substring(int i, int j))` – המחזירה את התת-מחרוזת של `s` המתחילה במקומות `i` ועד לסוף המחרוזת.
- `(j .. s.substring(int i, int j))` – המחזירה את התת-מחרוזת של `s` המתחילה במקומות `i` ועד למיקום `j-1`.

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

בשני הסעיפים:

- אפשר להשתמש בהעמסת-יתר (overloading).
- אפשר להניח שהמחרוזות אינן `null` ואין ריקות. אין צורך לבדוק זאת!
- אין צורך לדאוג ליעילות השיטה! אבל כמובן צריך לשימוש לב לא לעשות קריאות רקורסיביות מיותרות!
- אל תשכחו לטעוד את מה שכתבתם!



**שאלה 2 (25 נקודות)**

נתון מערך חד ממדי a המלא במספרים שלמים חיוביים ושליליים ללא אפס! (אין צורך לבדוק זאת). המערך אינו ממוקן!

כתבו שיטה סטטית `longestSubarray` המקבלת את המערך כפרמטר ומחזירה את אורך של התת-מערך (הרציף) הארוך ביותר שכל איבריו הם לסירוגין מספר חיובי ומספר שלילי. השיטה צריכה גם להדפיס מה האינדקס התחלתי ומה האינדקס הסופי של התת-מערך המבוקש. אם יש במערך יותר מהתת-מערך אחד באורך מקסימלי, יודפסו האינדקסים של התת-מערך הראשון מביניהם.

**לדוגמא:**

- עבור המערך  $\{2, -1, 1, -1, -5, 2, 1\}$  השיטה **תחזיר 3** (האיברים של התת-מערך הזה מודגשים) ו**תדפיס** מהו כעין זה :

Starting index = 0 Ending index = 2

- עבור המערך  $\{-2, -1, 1, 2, -7, 2, 1, 3, 3\}$  השיטה **תחזיר 3** ו**תדפיס** מהו כעין זה :

Starting index = 2 Ending index = 4

- עבור המערך  $\{1, 2, 3, 4, 5, 4\}$  השיטה **תחזיר 1** (אין תת-מערך באורך גדול מ-1 שיש בו איברים מתחלפים. لكن כל אחד מהאיברים במערך הוא תת-מערך באורך 1) ו**תדפיס** מהו כעין זה :

Starting index = 0 Ending index = 0

- עבור המערך  $\{1, -2, 3, -4, -5, 4, 2, -4, 6, -2\}$  השיטה **תחזיר 4** (יש שני תת-מערכות באורך 4, אחד מאינדקס 0 עד אינדקס 3, והשני מאינדקס 6 עד אינדקס 9) ו**תדפיס** מהו כעין זה :

Starting index = 0 Ending index = 3

**חתימת השיטה היא:**

```
public static int longestSubarray (int[] a)
```

**מה סיבוכיות זמן הריצה והמקום של השיטה שתכתבם? הסבירו תשובהכם.**

**שימוש לב:**

השיטה שתכתבו צריכה להיות עיליה כלל הניתן, גם מבחינת סיבוכיות הזמן וגם מבחינת סיבוכיות המקום. תשובה שאינה עיליה מספיק כלומר, שתהייה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל מעט נקודות בלבד.

**אל תשכחו לתעד את מה שתכתבם!**



**חלק ב - את התשובות לשאלות 3- 5 יש לכתוב על גבי השאלה.  
לא נבדוק תשובות שייכתבו במקום אחר!**


**שאלה 3 (20 נקודות)**

נניח שהמחלקה Node שלහן מ眞מת עצ' בינו'.

```
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;}  
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעצ' בינו'.  
בין השיטות נתונות השיטות static what, what something,秘密 secret, secret מהן מקבלת שורש של  
עצ' בינו'.

```
public static int what(Node root)  
{  
    if (root == null)  
        return 0;  
    return what(root.getLeftSon()) + 1 +  
           what(root.getRightSon());  
}  
  
public static boolean something(Node node)  
{  
    if (node == null)  
        return true;  
    if (node.getNumber() <= 0)  
        return false;  
    return something(node.getLeftSon()) &&  
           something(node.getRightSon());  
}
```

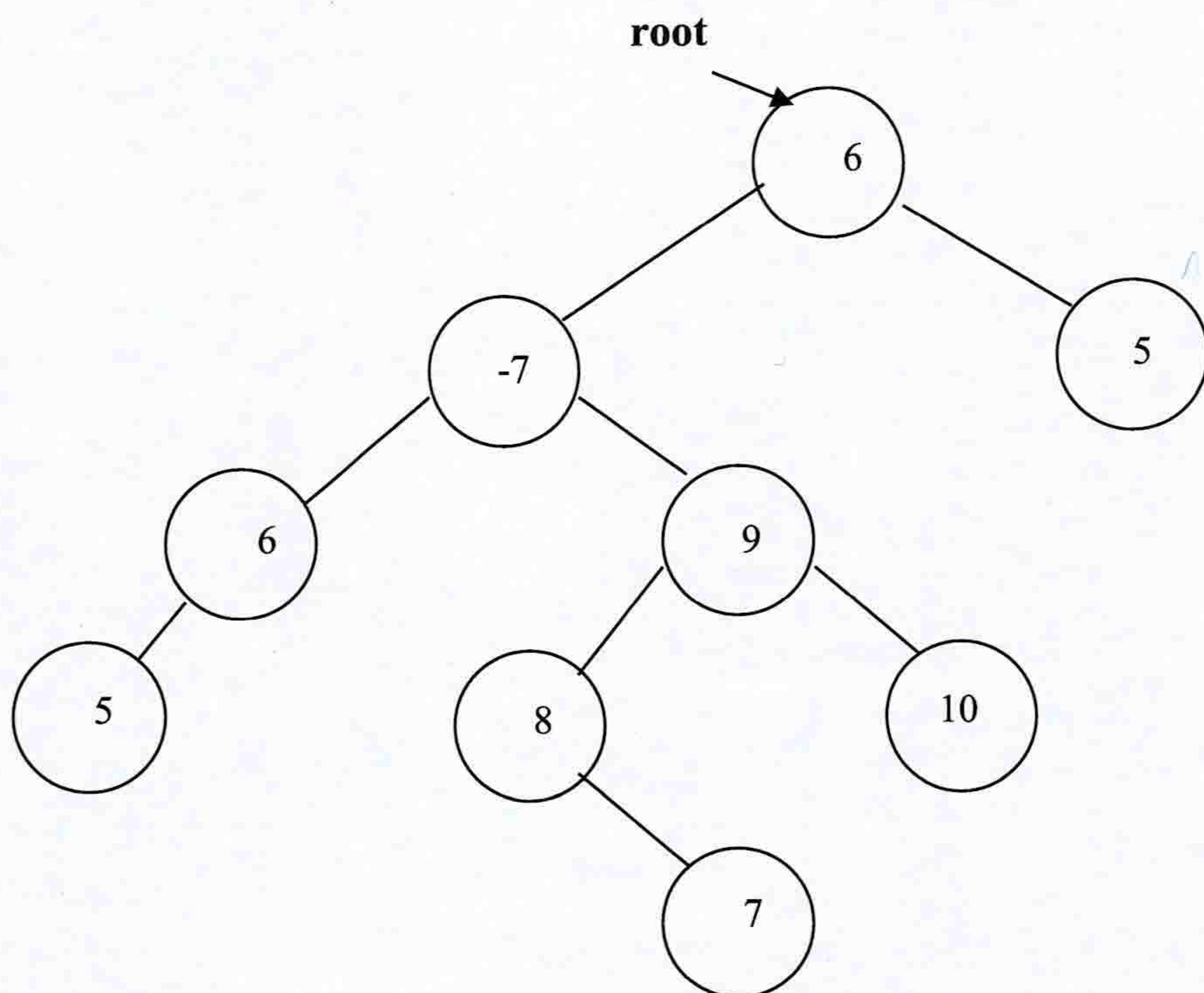


```

public static int secret(Node root)
{
    if (something(root))
        return what(root);
    return Math.max(secret(root.getLeftSon()),
                    secret(root.getRightSon()));
}

```

נתון העץ הבינרי הבא, ששורשו הוא root



root

5

6

−7

6

5

9

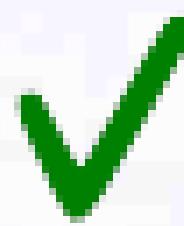
8

10

7

ענו על הטעיפים הבאים:

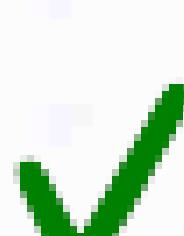
(2 נק') (א) איזה ערך תחזיר השיטה `what(root)` בעקבות הקראיה ?




התשובה היא:

(3 נק') (ב) מה מבצעת השיטה `what` באופן כללי כשהיא מקבל כפרמטר שורש של עץ ביני root? שימו לב, עלייכם לתת תיאור ממצה של מה עושה השיטה **באופן כללי**, ולא תיאור של מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. **כלומר**, מה המשמעות של הערך שהשיטה מחזירה? התיחסו ל McKee קצה!

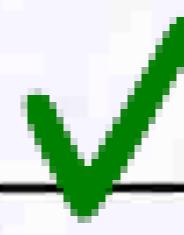
התשובה היא:





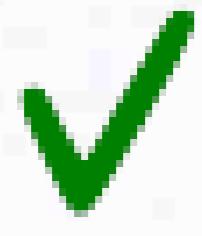
(3 נק') (ג) איזה ערך תחזיר השיטה **something** בעקבות הקריאה  
?BinaryTree.something(root)

התשובה היא:



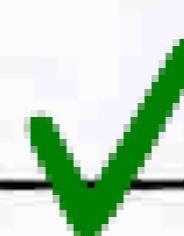
(4 נק') (ד) מה מבצעת השיטה **something** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בineri?  
שיםו לב, עלייכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של  
מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. **כלומר, מה המשמעות של הערך  
שהשיטה מחזירה? התיחסו למקרי קצה!**

התשובה היא:



(3 נק') (ה) איזה ערך תחזיר השיטה **secret** בעקבות הקריאה (root)?BinaryTree.secret(root)

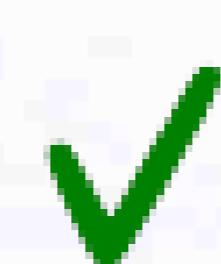
התשובה היא:



(5 נק') (ו) מה מבצעת השיטה **secret** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בineri?  
שיםו לב, עלייכם לתת תיאור ממצה של מה עושה השיטה באופן כללי, ולא תיאור של  
מה עושה כל שורה בשיטה, או איך היא מבצעת זאת. **כלומר, מה המשמעות של הערך  
שהשיטה מחזירה? התיחסו למקרי קצה!**

התשובה היא:

**ניסוח לקיי**



20  
(3)



שאלה 4 (14 נקודות)

נתון פרויקט שהוגדרו בו המחלקות שלהלן. כל אחת בקובץ נפרד, כמובן.

```
public class Animal
{
    public Animal()
    {
        System.out.println("new Animal");
    }
}

-----
public class Dog extends Animal
{
    protected String _name;
    public Dog(String n)
    {
        _name = n;
        System.out.println("new Dog");
    }
    public void printName()
    {
        System.out.println(_name);
    }
    public void compare(Dog d)
    {
        if (d._name.equals(_name))
            System.out.println("Dog - Yes");
        else
            System.out.println("Dog - No");
    }
}

-----
public class Poodle extends Dog
{
    public Poodle()
    {
        super("Kiki");
        System.out.println("new Poodle");
    }

    public void compare(Dog a)
    {
        System.out.print("Comparing... ");
        compare((Animal) a);
    }

    public void compare(Animal a)
    {
        if ((a instanceof Poodle) ||
            (a instanceof Dog) && ((Dog)a)._name.equals("Kiki"))
            System.out.println("Poodle - Positive");
        else
            System.out.println("Poodle - Negative");
    }
}
```



```

public class Dalmatian extends Dog
{
    public Dalmatian()
    {
        super("Kiki");
        System.out.println("new Dalmatian");
    }

    public void compare(Dog a) {
        System.out.print("Comparing...");
        compare((Animal) a);
    }

    public void compare(Poodle a) {
        System.out.print("Comparing Poodle...");
        compare((Animal) a);
    }

    public void compare(Animal a) {
        if ((a instanceof Poodle) ||
            (a instanceof Dog) && ((Dog)a).name.equals("Kiki"))
            System.out.println("Dalmatian Poodle - Positive");
        else
            System.out.println("Dalmatian Poodle - Negative");
    }

    public void compare(Dalmatian a) {
        if ((a instanceof Dalmatian) ||
            (a instanceof Dog) && ((Dog)a).name.equals("Kiki"))
            System.out.println("Dalmatian Dal - Positive");
        else
            System.out.println("Dalmatian Dal - Negative");
    }
}

```

כמו כן נתונה המחלקה Driver (באותו פרויקט) :

```

public class Driver
{
    public static void main(String [] args)
    {
        Animal a = new Animal();
        Poodle p1 = new Poodle();
        Dog d = new Dalmatian ();
        Animal d2= new Dog("Kiki");
        Dog d3 = new Dog("Max");
        Animal p2 = new Poodle();
        Dog p3 = new Poodle();
        Dalmatian d4 = new Dalmatian();

        כאן יופיעו הפקודות שבסעיפים הבאים //
    }
}

```

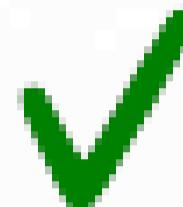
תְּאַמְּנָה לְכָלֶבֶת תְּשִׁיבֵךְ



בסעיפים הבאים נוסיף פקודות לשיטה main במחלקה Driver. בכל סעיף הניחו שהפקודות החדשות נוספו ללאו שכבר הוספו בסעיפים הקודמים לה. כל סעיף 2 פקודות.  
**אין צורך לכתוב את מה שיודפס בעקבות הצהרות האובייקטים והקריאות לבנאים.**  
 בכל סעיף כתבו מה יהיה הפלט שיודפס בעקבות הוספת הפקודות החדשות. (התיחסו לפט של הפקודות החדשות בלבד, אין צורך לרשום שוב ושוב את הפלט של הפקודות הקודמות). אם לדעתכם לא יודפס דבר, או תהיה שגיאה כלשהי – ציינו את השגיאה שנגרמה (קומפילציה/ריצחה) וסבירו את הסיבה לשגיאת.

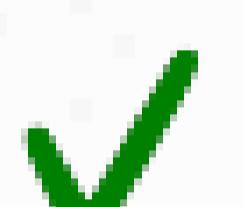
1. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(a);



2. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(d2);



3. מה יקרה בעקבות הוספת הפקודה הבאה?

d3.compare(p1);



4. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(d4);



5. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(p1);



6. מה יקרה בעקבות הוספת הפקודה הבאה?

((Dog)p2).compare(p3);



7. מה יקרה בעקבות הוספת הפקודה הבאה?

((Dalmatian)d).compare(a);

2  
4

לَا لَا לְכַתּוּ בָשָׁלִים


## שאלה 5 (16 נקודות)

נתונה המחלקה Queue המממשת תור, המאחסן מספרים שלמים.

להלן ה-API של המחלקה Queue :

Constructor Summary	
Constructors	Constructor and Description
	<code>Queue ()</code> Constructs an empty Queue object.
Method Summary	
All Methods	Instance Methods
Modifier and Type	Method and Description
<code>int</code>	<code>dequeue ()</code> Removes an item from the front of the queue.
<code>void</code>	<code>enqueue (int num)</code> Adds an item to the tail of the queue.
<code>boolean</code>	<code>isEmpty ()</code> Checks whether the queue is empty.

נניח שנוספה למחלקה גם השיטה `length()` שמחזירה את אורךו של התור (כמה איברים יש בו).  
הניחו שהשיטות לעיל ממומשות במחלקה Queue.  
במחלקה Queue הוגדרה גם השיטה הסטטית `what` שמתווארת בעמוד הבא. השיטה מקבלת כפרמטר תור `q` שמלא במספרים שלמים ומחזירה את התור `q`.



```

public static Queue what(Queue q)
{
    int length = q.size();
    Queue q1 = new Queue();
    for (int i=0; i<length/2; i++)
    {
        int num = q.dequeue();
        q1.enqueue(num);
    }
    Queue q2 = new Queue();
    for (int i=0; i<length/2; i++)
    {
        int num = q1.dequeue();
        q2.enqueue(num);
        num = q.dequeue();
        q2.enqueue(num);
    }
    for (int i=0; i<length; i++)
    {
        int num = q2.dequeue();
        q.enqueue(num);
    }
    return q;
}

```

**סעיף א (4 נק')**

אם נפעיל את השיטה **what** על התור {1, 2, 3, 4, 5, 6} כאשר 1 נמצא בראש התור ו- 6 בזנבו איך ייראה התור פ המוחזר?

**התשובה היא**

**סעיף ב (4 נק')**

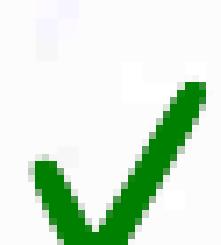
לאחר הפעלת השיטה **what** עם הפרמטר שהוא תור פ מלא במספרים שלמים, הוחזר התור פ הבא:

{8, 4, 7, 3, 6, 2, 5, 1}

(כasher 8 נמצא בראש התור ו- 1 בזנבו).

מה היה התור פ שהועבר כפרמטר לשיטה?

**התשובה היא**





**סעיף ג (4 נק')**

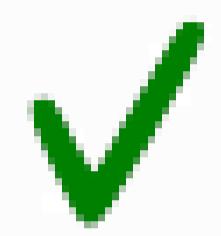
לאחר הפעלת השיטה **what** עם הפרמטר שהוא תור **q** המלא במספרים שלמים, הוחזר התור **q** הבא:

{1, 0, 0, 1, 1, 0}

(כאשר 1 נמצא בראש התור ו- 0 בזנבו).

מה היה התור **q** שהועבר כפרמטר לשיטה?

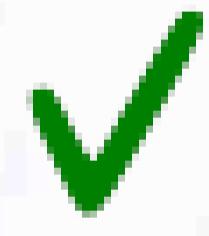
**התשובה היא**



**סעיף ד (4 נק')**

הציגו תור ובו לפחות 4 איברים שאם השיטה תופעל עליו - התוכנית תעוף בגלל שגיאת זמן ריצה, והסבירו מדוע השיטה עפה.

**התשובה היא**



**בהצלחה!**

**16  
(5)**

