

דף סיכום בבחינה

מזהה סטודנט: N101892923

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

מספר שאלה	ניקוד מירבי	צין	ניקוד
1.1	15.00	15.00	15.00
1.2	10.00	8.00	8.00
2	25.00	22.00	22.00
3	20.00	20.00	20.00
4	14.00	12.00	12.00
5	16.00	16.00	16.00

ציון בבחינה סופי : 93.00

הבחינה הבודקה בעמודים הבאים

5|3|5|8|9|2

מספר התלמיד הנבחן  
רשום את כל תשע הספרות

האוניברסיטה  
הפתוחה

כ' באדר א' תשפ"ב

סמסטר 2022א

20441 / 4

473 שאלון -

21

בפברואר 2022

89 מס' מועד

### שאלון בחינת גמר

## 20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 13 עמודים

### מבנה הבחינה:

- קראו בעיון את ההנחיות שלhallon:
  - \* בבחינה יש חמיש שאלות.
  - \* כל התכניות צrüיכות להיות מתועדות היטב.
  - יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית.
  - יש לבחור בשמות משמעותיים למשתנים, לפונקציות ולקבועים שבתוכנית.
  - תכנית שלא תתווד כנדרש לעיל תקבל לכל היותר % 85 מהניקוד.
  - \* יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
  - תכנית לא יעילה לא תקבל את מלאה הנקודות.
  - \* אם ברצונכם להשתמש בתשובותם בשיטה או בחלוקת הכתיבה בחוברת השקפים, אין צורך שתעתיקו את השיטה או את המחלוקת מהבחינה. מספיק להפנות למקום הנכון, וב惟ך שההפניה תהיה מדויקת (פרמטרים, מקום וכו').
  - \* אין להשתמש בחלוקת קיימות ב- Java, חוץ מآلומ הפורטות בשאלות הבחינה.
  - \* יש לשמר על סדר; תכנית הכתיבה בצורה בלתי מסודרת עלולה לגרום מהציוון.
  - \* בכתיבה התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה  
אין להשתמש במשתנים גלובליים!
  - \* אפשר לטע ב עברית. אין צורך בתיעוד API .
- כל התשובות צrüיכות להכתב בתווך קובץ המבחן במקומות המתאים בלבד.  
תשובה שתיכתב שלא במקומה לא תיבדק.

### חומר עזר:

חוברות השקפים 6-1, 12-7.

אין להכניס חומר מודפס נוספת או חומר אחר מכל סוג.

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

### החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתווך לחברת התשובות

בהצלחה !!!





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

### שאלה 1 (25 נקודות)

בhinתן 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 "adbrcfa" = 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 "adbfbaagcxabcd" = str1 פעמיים בדילוגים קבועים. פעם אחת בדילוגים של 4 (החל מהתו הראשון במחרוזת str1), ופעם שנייה בדילוגים של 1 (מיד אחרי התו '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). **נתיחה רק למחרוזות שמתחלות באותו تو.**

### סעיף א: (15 נקודות)

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

לדוגמא:

1. `str1 = "adbrcfa", str2 = "abc"` step = 2 אם יוחזר true .
2. `str1 = "cbdadbrcfa", str2 = "abc"` step = 2 אם יוחזר false למרות שהמחרוזת str2 מופיעה בדילוג קבוע של 2 בתוך str1, אבל הדילוג לא מתחיל בתו הראשון של str1.
3. `str1 = "adcfbaagcxabcd", str2 = "abc"` step = 1 אם יוחזר true , אבל אם step = 4 יוחזר false כי הדילוג הזה לא מתחיל בתו הראשון של str1.
4. `str1 = "abc", str2 = "abc"` step = 4 ואם אם step = 1 יוחזר true , וגם אם step = 4 יוחזר true . כי שני הדילוגים מתחלים בתו הראשון של str1.

חתימת השיטה היא:

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

```
if(str1.length() >= str2.length()) {  
    int temp = strStep(str1, str2, str1.length());  
    if (temp == Integer.MAX_VALUE)  
        return -1;  
    return temp;  
} else {  
    return -1;  
}
```

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

לדוגמא, בדוגמאות של סעיף א –

- 1. השיטה `strStep(str1, str2 = "adbrfa", str2 = "abc")` תחזיר את הערך `2`.
- 2. השיטה `strStep(str1, str1 = "cbdadabrcfa", str2 = "abc")` תחזיר את הערך `-1`.
- 3. השיטה `strStep(str1, str1 = "adcfbaagcxabcd", str2 = "abc")` תחזיר את הערך `4`.
- 4. השיטה `strStep(str1, str1 = "abcfbaagcxabcd", 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)
```

שים לב, מותר לכם להשתמש בשיטה `JumpJS` 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 התוו הנמצא במקומות `index` במחרוזת `s`
- `s.indexOf(char c)` – המחזירה את המיקום הראשון בו נמצא התוו `c` במחרוזת `s`
- `s.substring(int i)` – 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`.

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 בהעמתה-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 קרייאות רקורסיביות מיותרות!
- אל תשכחו לטעד את מה שתכתבם!



## שאלה 2 (25 נקודות)

נתון מערך חד ממדי a המלא במספרים שלמים חיוביים ושליליים **לא 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 מתת-מערך אחד באורך מקסימלי, יודפסו האינדקסים של התת-מערך הראשון מביניהם.

לדוגמא:

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

Starting index = 0 Ending index = 2

$\begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix}$  השיטה תחזיר 3

- עבור המערך  $\{ -2, -2, 1, 1, 2, -7, 2, 1, 1, 2, 3, 3 \}$  השיטה תחזיר 3 ותדפיס משוה כעין זה:

Starting index = 2 Ending index = 4

- עבור המערך  $\{ 1, 2, 3, 4, 5, 4, 1, 2, 3, 4, 5, 6 \}$  השיטה תחזיר 1 (אין תת-מערך באורך גדול מ-1 שיש בו איברים מתחלפים. לכן כל אחד מהאיברים במערך הוא תת-מערך באורך 1) ותדפיס משוה 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) ותדפיס משוה כעין זה:

Starting index = 0 Ending index = 3

חותמת 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 הכל 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 תשבות שיכתבו במקום אחר!**

**שאלה 3 (20 נקודות)**

נניח שהמחלקה Node שלහלו מ眞מת עצם ביניי.

```
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 בעץ ביניי.  
בין השיטות נתונות השיטות what, something, what something ו- secret 받아ות, שכל אחת מהן מקבלת שורש של  
עץ ביניי.

```
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());  
}
```

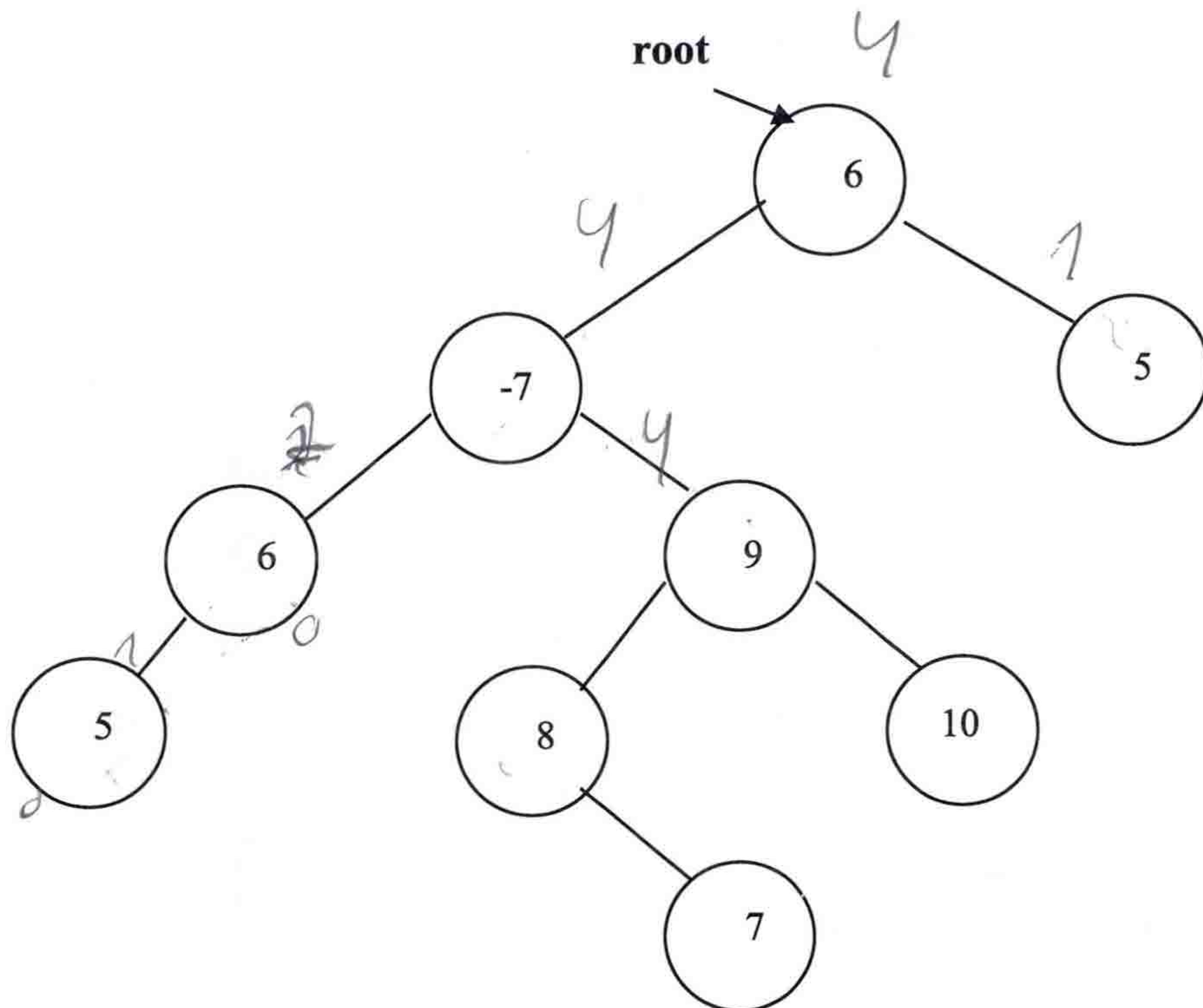


```

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

```

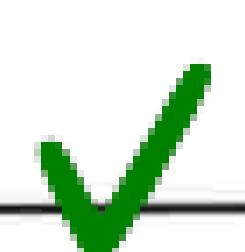
נתנו העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

(2 נקי) (א) איזה ערך תחזיר השיטה `what` בעקבות הקראיה `?BinaryTree.what(root)`

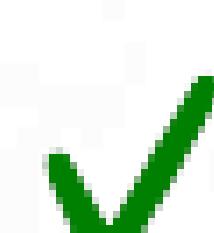
התשובה היא:



9

(3 נקי) (ב) מה מבצעת השיטה `what` באופן כללי כשהיא מקבל כפרמטר שורש של עץ בינירי `root`?  
שים לב, עליכם לתת תיאור ממצה של מה עושה השיטה `what` 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:  
השיטה `what` מקבל שורש העץ כפרמטר ומחזירה את המספר המרבי בין כל שורה.

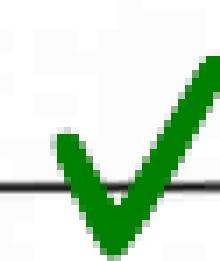




(3 נק') (ג) איזה ערך תחזיר השיטה **something** בעקבות הקריאה

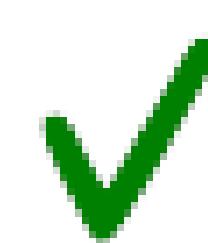
?BinaryTree.something(root)

התשובה היא:



false

(4 נק') (ד) מה מבצעת השיטה **something** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי? שימו לב, עליכם 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!**

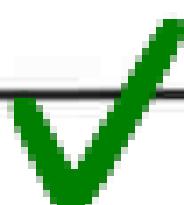


התשובה היא:

השיטה **something** מקבלת גזע העץ root ופונקציית **something** מודifies את root. true (אנט-פלס-truish) מודifies את root. false (אנט-פלס-false). אם root הוא true מוחזק root false. אם root הוא false מוחזק root true.

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

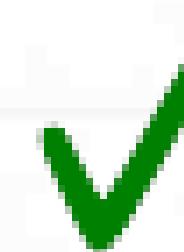
התשובה היא:



4

(5 נק') (ו) מה מבצעת השיטה **secret** באופן כללי כשהיא מקבלת כפרמטר שורש של עץ בינרי root? שימו לב, עליכם 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!**

השיטה **secret** מקבלת גזע העץ root ופונקציית **secret** מודifies את root. true (אנט-פלס-truish) מודifies את root. false (אנט-פלס-false). אם root הוא true מוחזק root false. אם root הוא false מוחזק root true.



20

(3)



שאלה 4 (14 נקודות)

נתון פרויקט שהוגדרו בו המחלקות שלහן. 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 נקבעה המחלקה Driver (באותנו 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
    }
}

```

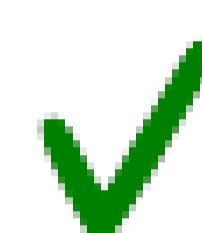


בסעיפים הבאים נווסף פקודות לשיטה main במחלקה Driver. בכל סעיף הניתן שהפקודות החדשות נוספות לאלה שכבר הוספו בסעיפים הקודמים לה. כל סעיף 2 פקודות.  
 אין צורך לכתוב את מה שיופיע בעקבות הצהרות האובייקטיבים והקריאות לבנאים.  
 בכל סעיף כתבו מה יהיה הפלט שיופיע בעקבות הוספת הפקודות החדשות. (התיחסו לפלט של 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/ריצה) והסבירו את הסיבה לשגיאה.

1. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(a);

Poodle - Negative



2. מה יקרה בעקבות הוספת הפקודה הבאה?

p1.compare(d2);

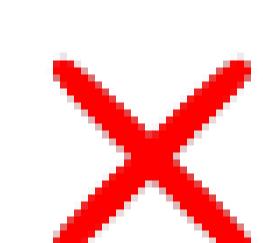
Poodle - Positive



3. מה יקרה בעקבות הוספת הפקודה הבאה?

d3.compare(p1);

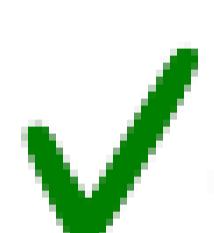
Poodle - Positive



4. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(d4);

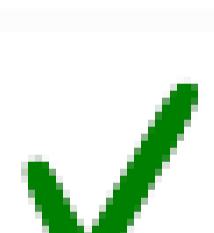
Comparing... Dalmatian Poodle - Negative



5. מה יקרה בעקבות הוספת הפקודה הבאה?

d.compare(p1);

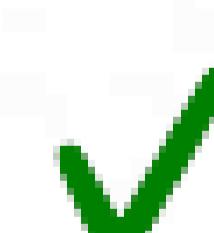
Comparing... Dalmatian Poodle - Positive



6. מה יקרה בעקבות הוספת הפקודה הבאה?

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

Comparing... Poodle - Positive



7. מה יקרה בעקבות הוספת הפקודה הבאה?

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

Dalmatian Poodle - Negative



12  
(4)



## 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	Concrete Methods
Modifier and Type	Method and Description	
int	<code>dequeue ()</code> Removes an item from the front of the queue.	
void	<code>enqueue (int num)</code> Adds an item to the tail of the queue.	
boolean	<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` שמתוארת 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;
}

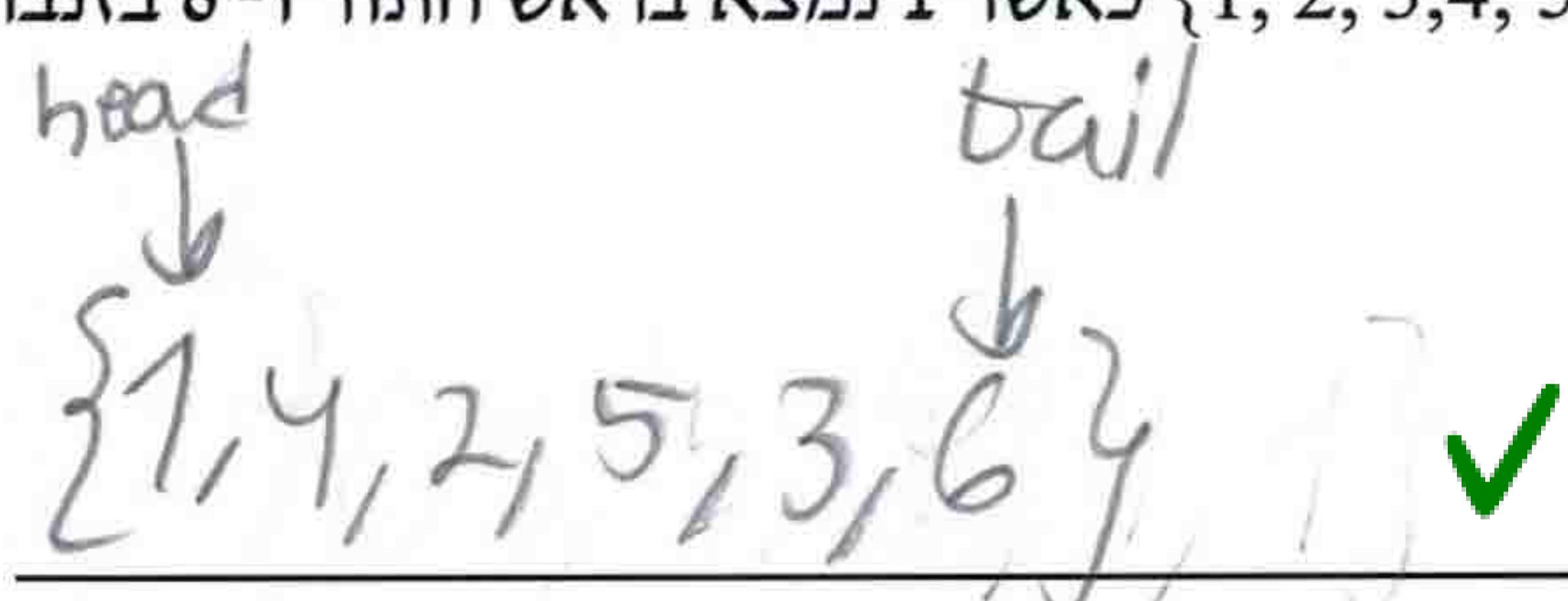
```

#### 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

יראה התור פ המוחזר?

**התשובה היא**



#### סעיף ב (4 נק')

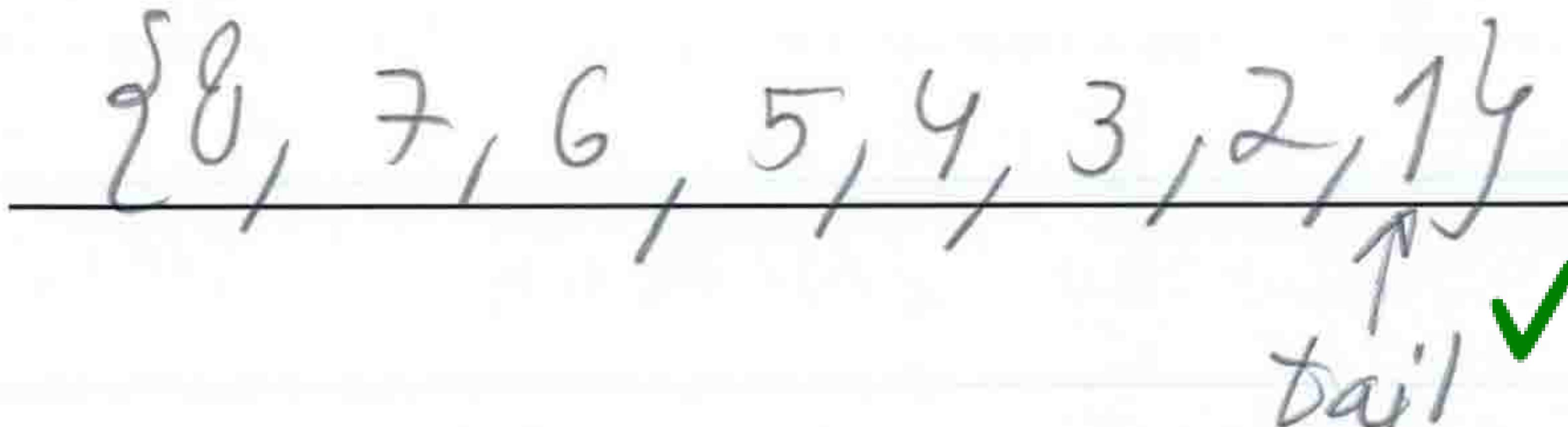
לאחר הפעלת השיטה **what** עם הפרמטר שהוא תור פ מלא במספרים שלמים, הוחזר התור פ הבא:

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

(כאשר 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 `շטח` תור `q` שמלא במספרים שלמים, הוחזר התור `q` הבא:

$$\{1, 0, 0, 1, 1, 0\}$$

(כארר 1 נמצא בראש התור ו- 0 בזנבו).

**מה היה התור ושהו עבר כפרמטר לשיטה?**

התשובה היא

סעיף ד' (4 נק')

הציגו תור ובו לפחות 4 איברים שם התוכנית תעוף בגל שגיאת זמן ריצה, וסבירו מדוע השיטה עפה.

ה问题是 מה הערך הראשון?  
head  
tail  
 $\downarrow$   
 $\{1, 2, 3, 4, 5\}$

הנתקה משלב הנתקה משלב הנתקה משלב הנתקה משלב

ב הצלחה!

1

16

6

