

מס' שאלון - 478  
בפברואר 2019

מס' מועד 84

סמסטר 2019א

20441 / 4

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 13 עמודים

מבנה הבחינה:

- קראו בעיון את ההנחיות שלהלן:
- \* בבחינה יש חמש שאלות.
- \* 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.
- \* יש להקפיד לכתוב את התכניות בצורה מבנית ויעילה.
- תכנית לא יעילה לא תקבל את מלוא הנקודות.
- \* אם ברצונכם להשתמש בתשובתכם בשיטה או במחלקה הכתובה בחוברת השקפים, אין צורך שתעתיקו 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.
- \* בכתיבת התכניות יש להשתמש אך ורק במרכיבי השפה שנלמדו בקורס זה
- אין להשתמש במשתנים גלובליים!
- \* את התשובות לשאלות 3 - 5 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!
- \* אפשר לתעד בעברית. אין צורך בתיעוד API.

חומר עזר:

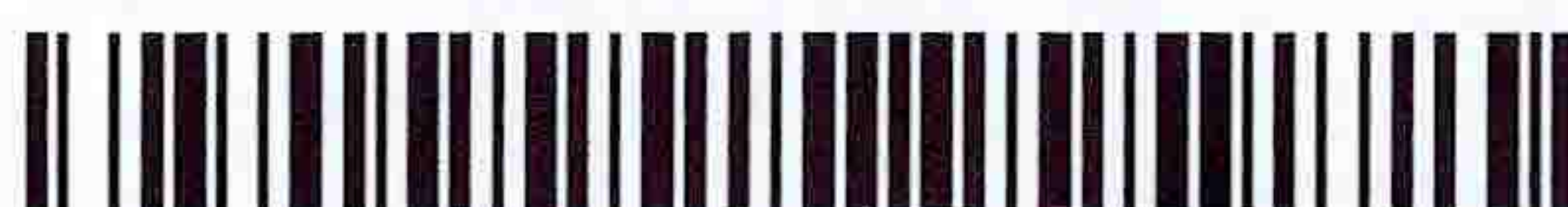
ספר הלימוד : java software solutions מאת : lewis/loftus  
חוברת השקפים של הקורס של ד"ר אמיר גורן ותמר וילנר.  
יחידות 1-6, 7-12. מותרות הערות בכתב יד, ע"ג הספרים.  
אין להכניס חומר מודפס או כל חומר אחר מכל סוג שהוא.

החזירו

למשגיח את השאלון

וכל עזר אחר שקיבלתם בתוך מחברת התשובות

בהצלחה !!!





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

שאלה 1 (25 נקודות)

כתבו שיטה סטטית בוליאנית רקורסיבית, המקבלת כפרמטר מספר שלם חיובי (ממש) num. השיטה צריכה לבדוק אם המספר הזה ניתן לכתיבה כסכום של חזקות של 3. כל חזקה של 3 יכולה להופיע בסכום לכל היותר פעם אחת. אם המספר ניתן לכתיבה כך, השיטה תחזיר את הערך true, ואחרת תחזיר false. לדוגמא:

- עבור  $num = 37$  יוחזר true כי:

$$3^0 + 3^2 + 3^3 = 1 + 9 + 27 = 37$$

- עבור  $num = 38$  יוחזר false כי אי אפשר לכתוב את 38 כסכום של חזקות של 3.

חתימת השיטה היא:

```
public static boolean sumPower3 (int num)
```

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).

אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות! אל תשכחו לתעד את מה שכתבתם!



## שאלה 2 (25 נקודות)

כתבו שיטה סטטית המקבלת כפרמטר מערך `arr` המכיל מספרים שלמים. השיטה אמורה להחזיר אינדקס `i` בעזרתו מחלקים את המערך לשני חלקים, שמאלי וימני, כאשר

- החלק השמאלי כולל את כל האיברים מתחילת המערך (אינדקס 0) עד `i` (כולל)
  - החלק הימני כולל את כל האיברים מאינדקס `i+1` ועד לסוף המערך (אינדקס `arr.length-1`)
- כך שמתקיים כי **ההפרש** (בערך מוחלט) בין ממוצע איברי החלק השמאלי של המערך, לבין ממוצע איברי החלק הימני של המערך הוא הגדול ביותר האפשרי.

### לדוגמא:

בהינתן המערך `arr` הבא:

0	1	2	3
5	7	-2	10

החלוקות האפשריות הן:

1.  $i=0$  החלק השמאלי הוא  $\{5\}$  והממוצע שלו 5, החלק הימני הוא  $\{7, -2, 10\}$  והממוצע שלו  $5 = (7+(-2)+10)/3$ . ההפרש הוא  $5 - 5 = 0$
2.  $i=1$  החלק השמאלי הוא  $\{5, 7\}$  והממוצע שלו  $6 = (5+7)/2$ , החלק הימני הוא  $\{-2, 10\}$  והממוצע שלו  $4 = ((-2)+10)/2$ . ההפרש הוא  $6 - 4 = 2$
3.  $i=2$  החלק השמאלי הוא  $\{5, 7, -2\}$  והממוצע שלו  $3.3333 = (5+7+(-2))/3$ , החלק הימני הוא  $\{10\}$  והממוצע שלו 10. ההפרש הוא  $10 - 3.333 = 6.666$

ולכן השיטה תחזיר 2 שזה האינדקס שמחלק את המערך כך שההפרש בין הממוצעים הוא מקסימלי. שימו לב שהשיטה לא מחזירה את ההפרש אלא את האינדקס.

אם קיים יותר מאינדקס אחד שנותן אותו הפרש מקסימלי, יש להחזיר אחד מהם באופן שרירותי.

חתימת השיטה היא:

```
public static int average (int [] arr)
```

### שימו לב:

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

מה סיבוכיות זמן הריצה של השיטה שכתבתם? הסבירו תשובתכם.  
אל תשכחו לתעד את מה שכתבתם!



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

שאלה 3 (15 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; }
}
```

המחלקה BinarySearchTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ חיפוש בינרי.

בין השיטות נתונות השיטות f, g ו- secret הבאות:

```
public static Node f(Node root)
{
    if (root==null)
        return null;
    if (root.getLeftSon() == null)
        return root;
    return f(root.getLeftSon());
}

public static Node g(Node root)
{
    if (root==null)
        return null;
    if (root.getRightSon() == null)
        return root;
    return g(root.getRightSon());
}
```



```

public static boolean secret(Node root, int num)
{
    Node a = f(root);
    Node b = g(root);
    return secret(root, root, num, a, b);
}

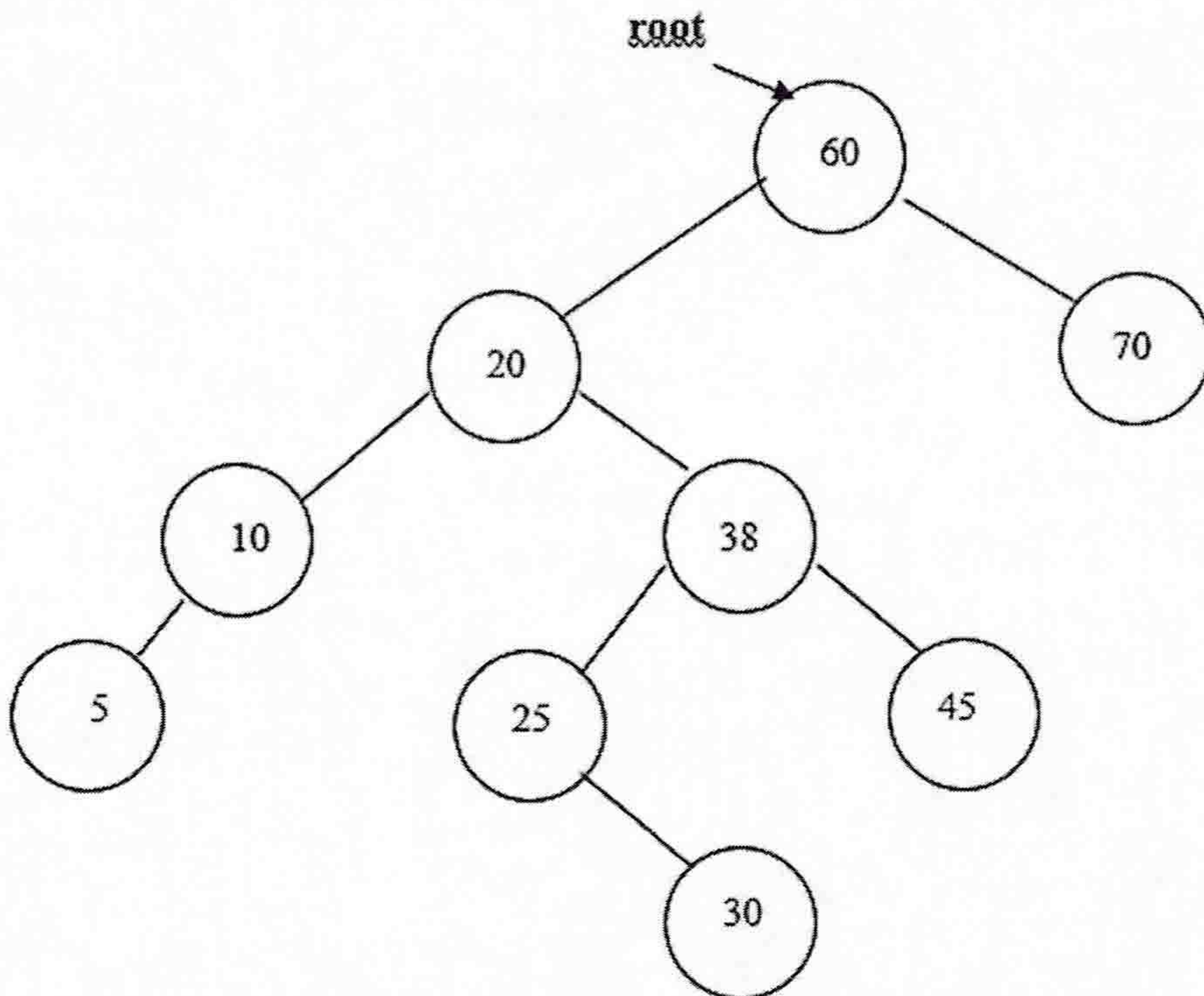
private static boolean secret(Node x, Node y, int num,
                               Node a, Node b)
{
    if (x == null || y == null)
        return false;

    if (a.getNumber() + b.getNumber() < num) ->
    {
        if (secret(x.getLeftSon(), y, num, a, b))
            return true;
        a = x;
        if (a != b && a.getNumber() + b.getNumber() == num)
        {
            System.out.print(a.getNumber()+" "+ b.getNumber());
            return true;
        }
        return secret(x.getRightSon(), y, num, a, b);
    }
    else
    {
        if (secret(x, y.getRightSon(), num, a, b))
            return true;
        b = y;
        if (a != b && a.getNumber() + b.getNumber() == num)
        {
            System.out.print(a.getNumber()+" "+ b.getNumber());
            return true;
        }
        return secret(x, y.getLeftSon(), num, a, b);
    }
}

```



נתון עץ החיפוש הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

2 נק' (i) מה תחזיר השיטה f בעקבות הקריאה BinarySearchTree.f(root)? **התשובה היא:**

✓ —

ומה תחזיר השיטה g בעקבות הקריאה BinarySearchTree.g(root)? **התשובה היא:**

✓ —

2 נק' (ii) מה מבצעת השיטה g באופן כללי כשהיא מקבלת כפרמטר שורש של עץ חיפוש

**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. שימו לב – כאן

**אנו שואלים על השיטה g ולא על f.**

**התשובה היא:**

✓ —  
—

3 נק' (iii) איזה ערך תחזיר השיטה secret ומה יודפס בעקבות הקריאה

BinarySearchTree.secret(root, 40)?

**התשובה היא:**

✓ —  
—  
—



3) (נק') (iv) איזה ערך תחזיר השיטה secret ומה יודפס בעקבות הקריאה

BinarySearchTree.secret (root, 60)?

התשובה היא:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

5) (נק') (v) מה מבצעת השיטה secret באופן כללי כשהיא מקבלת כפרמטרים שורש של עץ חיפוש בינרי root ומספר num כלשהו? 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.

התשובה היא:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

שאלה 4 (21 נקודות)

בהינתן המחלקות A, B ו-Driver הבאות (כל מחלקה נמצאת בקובץ נפרד):

```
public class A
{
    protected String s1;
    public A(){
        s1 = "abc";
    }

    public A(String s) { s1 = s; }

    public String get1() { return s1; }
}
//-----//
```



```

public class B extends A
{
    private String s2 = "ghi";
    public B(String s){
        s2 = s;
    }
    public B() {
        super("def");
    }

    public String get1() { return s1; }

    public String get2() { return s2; }
}

//-----//
public class Driver
{
    public static void main (String [] args)
    {
        // *** כאן ייכנסו השורות שבסעיפים להלן
    }
}

```

לכל אחד מהסעיפים שלהלן, כתבו מה יקרה בעקבות הוספתו לשיטה main שלעיל, (במקום שמסומן בכוכביות \*\*\*).

עבור כל שורה בסעיף כתבו- האם השורה לא עוברת קומפילציה? (כתבו למה). האם השורה עוברת קומפילציה אבל גורמת לשגיאת ריצה? (כתבו למה). האם השורה רצה בצורה תקינה? (כתבו מה הפלט מההרצה אם יש).

**כל סעיף מורץ בנפרד ולא תלוי בסעיפים האחרים!**

1. A a1 = new A();  
System.out.println(a1.get1());

**התשובה היא:**

שורה ראשונה

שורה שניה

✓  
\_\_\_\_\_  
\_\_\_\_\_

2. A a2 = new B();  
System.out.println(a2.get1());  
System.out.println(a2.get2());

**התשובה היא:**

שורה ראשונה

שורה שניה

שורה שלישית

✓  
\_\_\_\_\_  
✓  
\_\_\_\_\_  
\_\_\_\_\_



```
3. B b1 = new B("jkh");
   System.out.println(((A)b1).get1());
   System.out.println(((A)b1).get2());
   System.out.println(b1.get1());
   System.out.println(b1.get2());
```

התשובה היא:

- שורה ראשונה
- שורה שניה
- שורה שלישית
- שורה רביעית
- שורה חמישית

—  
—  
—

```
4. B b2 = (B)(new A("lmn"));
   System.out.println(b2.get1());
   System.out.println(b2.get2());
```

התשובה היא:

- שורה ראשונה
- שורה שניה
- שורה שלישית

```
5. A a3 = new B();
   B b3 = a3;
   System.out.println(b3.get());
```

התשובה היא:

- שורה ראשונה
- שורה שניה
- שורה שלישית

—  
—  
—

```
6. A a4 = new B("234");
   B b3 = (B)a4;
   A a5 = b3;
   System.out.println(a5.get1());
   System.out.println(a5.get2());
```

התשובה היא:

- שורה ראשונה
- שורה שניה
- שורה שלישית
- שורה רביעית
- שורה חמישית

—  
—  
—  
—  
—



נתונה המחלקה IntNode הבאה, המייצגת איבר ברשימה:

```
public class IntNode {
    private int _value;
    private IntNode _next;

    public IntNode(int val, IntNode n) {
        _value = val;
        _next = n;
    }

    public int getValue() {
        return _value;
    }
    public IntNode getNext() {
        return _next;
    }

    public void setValue(int v) {
        _value = v;
    }
    public void setNext(IntNode node) {
        _next = node;
    }
}
```

נתונה רשימה מקושרת של מספרים שלמים, הממומשת בעזרת המחלקה IntList שלהלן:

```
public class IntList
{
    private IntNode _head;

    public IntList( ) {
        _head = null;
    }

    public IntList(IntNode node) {
        _head = node;
    }

    public IntNode what()
    {
        IntNode result = null;
        IntNode current = _head;
        while (current != null)
        {
            IntNode next = current.getNext();
            current.setNext(result) ;
            result = current;
            current = next;
        }
        return result;
    }
}
```



```

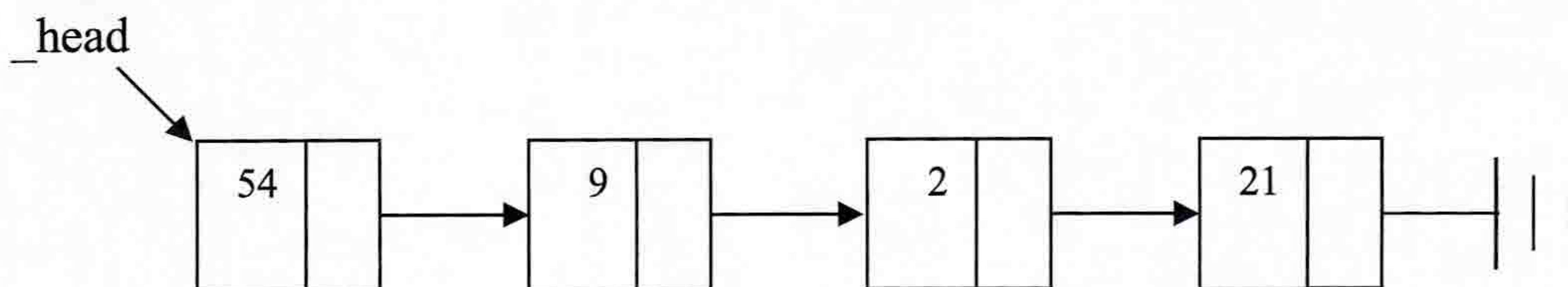
public IntNode secret()
{
    IntNode p = null;
    IntNode p1 = _head, p2 = _head;
    while (p2 != null && p2.getNext() != null)
    {
        p = p1;
        p1 = p1.getNext();
        p2 = p2.getNext().getNext();
    }
    if (p2 != null && p2.getNext() == null)
    {
        p = p1;
        p1 = p1.getNext();
    }
    p.setNext(null);
    return p1;
}

public boolean something()
{
    if (_head != null)
    {
        IntNode p = secret();
        IntList pList = new IntList(p);
        p = pList.what();
        IntNode p1 = _head;
        IntNode p2 = p;
        while (p1 != null && p2 != null)
        {
            if (p1.getValue() != p2.getValue())
                return false;
            p1 = p1.getNext();
            p2 = p2.getNext();
        }
        return true;
    }
    return true;
}

```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים.

להלן נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.  
 כך לדוגמא, נסמן { 54 , 9 , 2 , 21 } את הרשימה שלהלן:





ענו על הסעיפים הבאים :

**סעיף א (2 נקודות)**

איך תיראה הרשימה  $\{3, 6, 2, 7\} \rightarrow list1$  לאחר שנבצע את הפקודה הבאה :

```
list1 = new IntList (list1.what());
```

**התשובה היא:**

—  
—

**סעיף ב (2 נקודות)**

מה מבצעת השיטה `what` באופן כללי? הסבירו בקצרה מה השיטה עושה ולא כיצד היא מבצעת זאת.

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.

**התשובה היא:**

—  
—

**סעיף ג (2 נקודות)**

מה הערך אותו תחזיר השיטה `secret` כשנפעיל אותה על הרשימה  $\{3, 6, 2, 7\}$  ?

**התשובה היא:**

—

**סעיף ד (2 נקודות)**

מה הערך אותו תחזיר השיטה `something` כשנפעיל אותה על הרשימה  $\{3, 6, 2, 4, 6, 3\}$  ?

**התשובה היא:**

—

**סעיף ה (2 נקודות)**

כדי שהשיטה `something` תחזיר ערך אחר ממה שהחזירה בסעיף ה, מה השינוי המינימלי שצריך לעשות ברשימה  $\{3, 6, 2, 4, 6, 3\}$  (שינוי ברשימה פירושו, להחליף ערך של איבר לערך אחר, להוסיף איבר מסוים, להוריד איבר מסוים וכד').

**התשובה היא:**

—



#### סעיף ו (4 נקודות)

מה מבצעת השיטה **something** באופן כללי כשהיא מופעלת על רשימה כלשהי? הסבירו בקצרה מה השיטה עושה ולא **כיצד** היא מבצעת זאת.

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, עליכם לכתוב מה המשמעות של הערך שמוחזר מהשיטה **something**, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

### בהצלחה