

## קרא בעיון את ההנחיות שלהלן:

- במבחן יש חמש שאלות. עליכם לענות על כולן.
- כל התכניות צריכות להיות מתועדות היטב. יש לכתוב תחילה בקצרה את האלגוריתם וכל הסבר נוסף הדרוש להבנת התכנית. יש לבחור בשמות משמעותיים למשתנים, לשיטות 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>• את התשובות לשאלות 3 – 5 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!</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

**פלינדרום** (Palindrome) הוא מילה, מספר, משפט או כל רצף סמלים אחר שקריאתו מימין לשמאל ומשמאל לימין היא זהה.  
לדוגמא: מחרוזות התווים "aba", "1221" הן פלינדרום.

במערך חד-ממדי המכיל מספרים שלמים, נגדיר **רצף פלינדרומי** כסדרה של תאים רצופים במערך המהווים פלינדרום.

**לדוגמא:**

במערך arr הבא:

0	1	2	3	4	5	6	7	8
1	3	2	3	10	10	3	2	4

ישנם ארבעה רצפים פלינדרומיים:

1. בין האינדקסים 4 ל-5 {10, 10} אורכו 2
  2. בין האינדקסים 1 ל-3 {3, 2, 3} אורכו 3
  3. בין האינדקסים 3 ל-6 {3, 10, 10, 3} אורכו 4
  4. בין האינדקסים 2 ל-7 {2, 3, 10, 10, 3, 2} אורכו 6
- ועוד תשעה רצפים פלינדרומיים, כל אחד מהם באורך 1, שהם כל אחד מהתאים.

כתבו שיטה **סטטית רקורסיבית**, המקבלת כפרמטר מערך חד-ממדי arr המכיל מספרים שלמים ומחזירה את אורכו של הרצף הפלינדרומי הגדול ביותר.

לדוגמא, במערך arr שלעיל, הרצף הפלינדרומי הגדול ביותר הוא בין האינדקסים 2 ל-7, ואורכו הוא 6. לכן, השיטה צריכה להחזיר את הערך 6.  
על המערך  $arr = \{1, 2, 3, 4\}$  השיטה תחזיר 1.

**חתימת השיטה היא:**

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

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

**סדרה חשבונית (arithmetic sequence)** היא סדרה של מספרים, שבה ההפרש בין כל שני איברים עוקבים הוא קבוע.

לדוגמה: בסדרה  $3, 5, 7, 9, 11, \dots$  : ההפרש בין כל שני איברים עוקבים הוא קבוע – 2.

נתון מערך חד-ממדי arr מלא במספרים שלמים המהווים סדרה חשבונית. במערך נפלה טעות, ואחד המספרים חסר.

כתבו שיטה סטטית, המקבלת מערך arr חד-ממדי המכיל סדרה חשבונית, ומחזיר מיהו האיבר החסר.

**לדוגמא**, במערך arr הבא :

0	1	2	3	4	5
7	10	13	16	22	25

חסר המספר 19. לכן השיטה תחזיר את הערך 19.

**חתימת השיטה היא:**

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

אפשר להניח שמערך arr אכן מכיל סדרה חשבונית **ובדיוק** מספר אחד חסר בסדרה. אין צורך לבדוק זאת.

**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 secret ו-what הבאות:

```
public static Node secret(Node root, int num1, int num2)
//Assuming num1 <= num2
{
    if (root == null)
        return null;

    root.setLeftSon(secret(root.getLeftSon(), num1, num2));
    root.setRightSon(secret(root.getRightSon(), num1, num2));

    if (root.getNumber() < num1)
        root = root.getRightSon();

    else if (root.getNumber() > num2)
        root = root.getLeftSon();

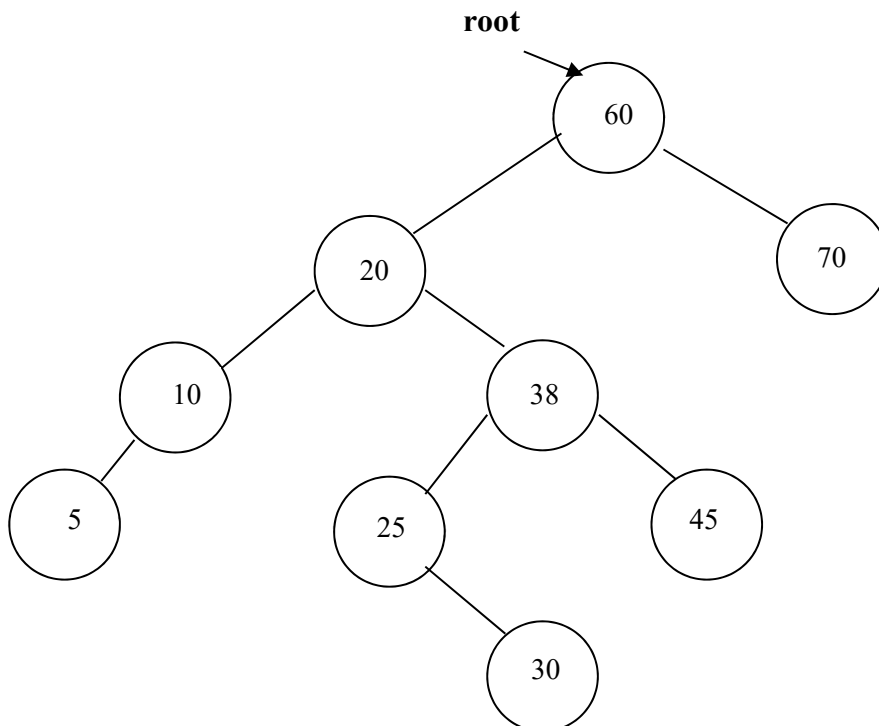
    return root;
}
```

```

public static void what(Node root)
{
    if (root != null)
    {
        what(root.getLeftSon());
        System.out.print (root.getNumber() + " ");
        what(root.getRightSon());
    }
}

```

נתון עץ החיפוש הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

(2 נק') (i) מה תדפיס השיטה **what** בעקבות הקריאה `?BinarySearchTree.what(root)`

התשובה היא:

---



---

(2 נק') (ii) מה מבצעת השיטה **what** 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.

התשובה היא:

---



---

4) (נק') (iii) איך ייראה העץ שמצויר לעיל ששורשו root לאחר שנפעיל את השיטה secret עם הפרמטרים שלהלן?

BinarySearchTree.secret (root, 10, 30)?

**התשובה היא:**

2) (נק') (iv) מה תדפיס השיטה **what** בעקבות הקריאה BinarySearchTree.what(root) **לאחר** שהופעלה עליו השיטה secret לעיל ?

**התשובה היא:**

---

---

5) (נק') (v) מה מבצעת השיטה secret באופן כללי כשהיא מקבלת כפרמטר שורש של עץ חיפוש בינרי root ושני מספרים שלמים num1 ו- num2, אפשר להניח ש-  $num1 \leq num2$ ? 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 {
    private String str1;

    public A() {
        this.str1 = "Hello";
        System.out.println("in A's " +
            "constructor " + str1);
    }

    public String getStr1() {
        return str1;
    }

    public void h() {
        System.out.println("in A's h");
    }

    public void f(){
        System.out.println("in A's f");
    }

    public void g() {
        System.out.println("in A's g");
    }

    public String toString() {
        return str1;
    }
}
//-----//
```

```

public class B extends A{
    private String str2;

    public B() {
        str2 = "Good Job!";
        System.out.println("in B's" +
            "constructor\n" + str2);
    }

    public B(A a) {
        if(a instanceof B){
            str2 = ((B) a).str2;
            System.out.println(str2);
            a.f();
        }
        else
            str2 = "Done";
    }

    public void h() {
        super.h();
        System.out.println("in B's h");
    }

    public void f() {
        System.out.println("in B's f");
    }

    public String toString() {
        return str2;
    }
}

//-----//
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 \*\*\*).

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

**שימו לב שלא בהכרח כל השורות יתמלאו**



```
1. A a1 = new A();  
   a1.h();
```

**התשובה היא:**

---

---

---

---

```
2. A a2 = new B();  
   a2.g();  
   a2.h();  
   a2.f();  
   B b1 = (B)a2;  
   b1.g();
```

**התשובה היא:**

---

---

---

---

---

---

---

---

```
3. A a2 = new B();  
   B b1 = (B)a2;  
   A a3 = new B(b1);  
   if (a3 instanceof B)  
       System.out.println ("Almost done");  
   else  
       System.out.println ("Just starting");
```

**התשובה היא:**

---

---

---

---

---

---

---

---

---

```
4. A m1 = new B(new A());  
   m1.h();
```

**התשובה היא:**

---

---

---

---

```
5. A a4 = new A();  
   B b2 = new B(a4);  
   System.out.println(b2);
```

**התשובה היא:**

---

---

---

```
6. B b3 = new B();  
   A a5 = b3;  
   Object obj1 = a5;  
   System.out.println(obj1);
```

**התשובה היא:**

---

---

---

---

## שאלה 5 - 14 נקודות

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 IntList what(int k)
    {
        IntNode x, y, prevX = null, prevY = _head;
        IntNode curr = _head;
        for (int i = 1; i < k && curr != null; i++)
        {
            prevX = curr;
            curr = curr.getNext();
        }
        x = curr;
    }
}
```

// המשך השיטה בעמוד הבא

```

    if (curr == null)
        return null;

    IntNode ptr = _head;
    while (curr.getNext() != null)
    {
        prevY = ptr;
        ptr = ptr.getNext();
        curr = curr.getNext();
    }
    y = ptr;

    if (x.getNext() == y)
    {
        x.setNext(y.getNext());
        y.setNext(x);
        prevX.setNext(y);
    }

    else if (y.getNext() == x)
    {
        y.setNext(x.getNext());
        x.setNext(y);
        prevY.setNext(x);
    }

    else if (x == _head)
    {
        _head = y;
        y.setNext(x.getNext());
        prevY.setNext(x);
        x.setNext(null);
    }

    else if (y == _head)
    {
        _head = x;
        x.setNext(y.getNext());
        prevX.setNext(y);
        y.setNext(null);
    }

    else {
        ptr = y.getNext();
        y.setNext(x.getNext());
        x.setNext(ptr);

        prevX.setNext(y);
        prevY.setNext(x);
    }

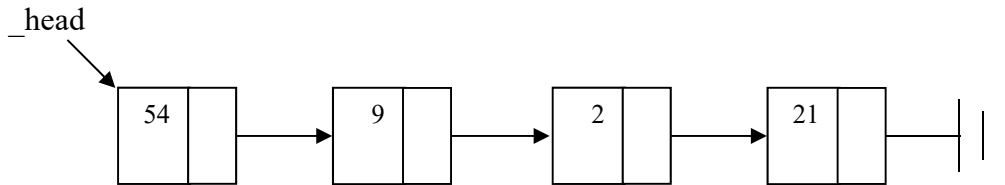
    return new IntList(_head);
}

```

## 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  $\{2, 17, 13, 7, 10, 5, 4, 20\}$   $\rightarrow$  list1 לאחר שנבצע את הפקודה הבאה:

```
list1 = new IntList (list1.what(3));
```

התשובה היא:

### סעיף ב (2 נקודות)

איך תיראה הרשימה list1 לאחר ביצוע סעיף א, לאחר שנבצע את הפקודה הבאה:

```
list1 = new IntList (list1.what(2));
```

התשובה היא:

### סעיף ג (10 נקודות)

מה מבצעת השיטה what באופן כללי כשהיא מופעלת על רשימה כלשהי ומקבלת מספר שלם

כלשהו num? 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 what, כשהיא מופעלת על רשימה כלשהי. התייחסו למקרי קצה.

התשובה היא:

## בהצלחה