

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

שאלה 1 (25 נקודות)

נתון מערך דו-ממדי `mat`, שערכיו הם המספרים השלמים 1, 0 או -1. מספר השורות לא בהכרח שווה למספר העמודות. נגדיר **מסלול** במערך אם הוא מקיים את התנאים הבאים:

1. התא הראשון במסלול הוא `mat[0][0]` (שורה ראשונה ועמודה ראשונה)
2. המסלול עובר אך ורק בתאים שמכילים 1 או 0.
3. בשורה עם אינדקס זוגי, אפשר ללכת אך ורק לתא מימין באותה שורה או לרדת לשורה הבאה באותה עמודה.
4. בשורה עם אינדקס אי-זוגי, אפשר ללכת אך ורק לתא משמאל באותה שורה או לרדת לשורה הבאה באותה עמודה.

עליכם לכתוב שיטה סטטית רקורסיבית המקבלת מערך דו-ממדי כנ"ל, המחזירה את סכום הנקודות הגבוה ביותר אליו אפשר להגיע במסלול כלשהו במערך הדו ממדי `mat`.

לדוגמא, מספר הנקודות המקסימלי במערך משמאל הוא 9. במערך מימין מסומן המסלול, ומודגשים התאים במסלול שערכם הוא 1.

| | 0 | 1 | 2 | 3 | 4 |
|---|----|----|----|----|----|
| 0 | 1 | 1 | -1 | 1 | 1 |
| 1 | 1 | 0 | 0 | -1 | 1 |
| 2 | 1 | 1 | 1 | 1 | -1 |
| 3 | -1 | -1 | 1 | 1 | 1 |
| 4 | 1 | 1 | -1 | -1 | 1 |

| | 0 | 1 | 2 | 3 | 4 |
|---|----|----|----|----|----|
| 0 | 1 | 1 | -1 | 1 | 1 |
| 1 | 1 | 0 | 0 | -1 | 1 |
| 2 | 1 | 1 | 1 | 1 | -1 |
| 3 | -1 | -1 | 1 | 1 | 1 |
| 4 | 1 | 1 | -1 | -1 | 1 |

אם בתא `[0][0]` הערך הוא -1 השיטה תחזיר -1. **חתימת השיטה היא:**

```
public static int findMaximum(int[][] mat)
```

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).

מוותר לשנות את המערך במהלך השיטה, אבל בסופה הוא צריך לחזור למצבו המקורי. אין צורך לדאוג ליעילות השיטה, אבל כמובן שצריך לשים לב לא לעשות קריאות רקורסיביות מיותרות! אל תשכחו לתעד את מה שכתבתם!


```
public static int findMaximum(int [] [] mat) {  
    return findMaximum(mat,0,0,0);  
}  
  
private static int findMaximum(int [] [] mat,int i,int j,int sum) {  
    if(mat[0][0] == -1)  
        return -1;  
    if(i < 0 || j < 0 || i >= mat.length || j >= mat[0].length || mat[i][j] == -1)  
        return sum;  
  
    int down = findMaximum(mat,i+1,j,sum+mat[i][j]);  
    int leftOrRight;  
  
    if(i % 2 == 0) {  
        leftOrRight = findMaximum(mat,i,j+1,sum+mat[i][j]);  
    }else {  
        leftOrRight = findMaximum(mat,i,j-1,sum+mat[i][j]);  
    }  
  
    return Math.max(down, leftOrRight);  
}
```


שאלה 2 (25 נקודות)

נתון מערך חד-ממדי a המלא במספרים שלמים.

כתבו שיטה סטטית, המקבלת כפרמטר מערך כזה, ומחזירה את מספר התת-מערכים בתוכו שהם מסודרים בסדר עולה ממש (כלומר ללא מספרים שווים). שימו לב שאנו מדברים על תאים סמוכים אחד לשני. אורך של תת-מערך מסודר בסדר עולה ממש הוא לפחות 2.

לדוגמא,

- אם נתון המערך a הבא :

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| a | = | 1 | 2 | 4 | 4 | 5 |

השיטה צריכה להחזיר את הערך 4.

אלו ארבעת התת-מערכים המסודרים בסדר עולה ממש :

{1, 2}, {1, 2, 4}, {2, 4}, {4, 5}

- אם נתון המערך a הבא :

| | | | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| a | = | 1 | 3 | 2 |

השיטה צריכה להחזיר את הערך 1. התת-מערך {1, 3}

- אם נתון המערך a הבא :

| | | | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| a | = | 5 | 4 | 3 | 2 | 1 |

השיטה צריכה להחזיר את הערך 0.

חתימת השיטה היא:

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

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 וסיבוכיות המקום של השיטה שכתבתם?
הסבירו תשובתכם. אל תשכחו לתעד את מה שכתבתם!


```
public static int strictlyIncreasing(int [] a) {  
    int sum = 0;  
    int counter = 0;  
    for(int i =0;i < a.length-1;i++) {  
        if(a[i] < a[i+1]) {  
            counter++;  
            sum += counter;  
        }else {  
            counter = 0;  
        }  
    }  
    return sum;  
}
```


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

שאלה 3 (17 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 SearchTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.

בין השיטות נתונות השיטות what, something ו- secret הבאות:

```
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, int min, int max)
{
    if (node == null)
        return true;
    if (node.getNumber() < min || node.getNumber() >= max)
        return false;
    return something(node.getLeftSon(), min, node.getNumber())
        && something(node.getRightSon(), node.getNumber(), max);
}
```

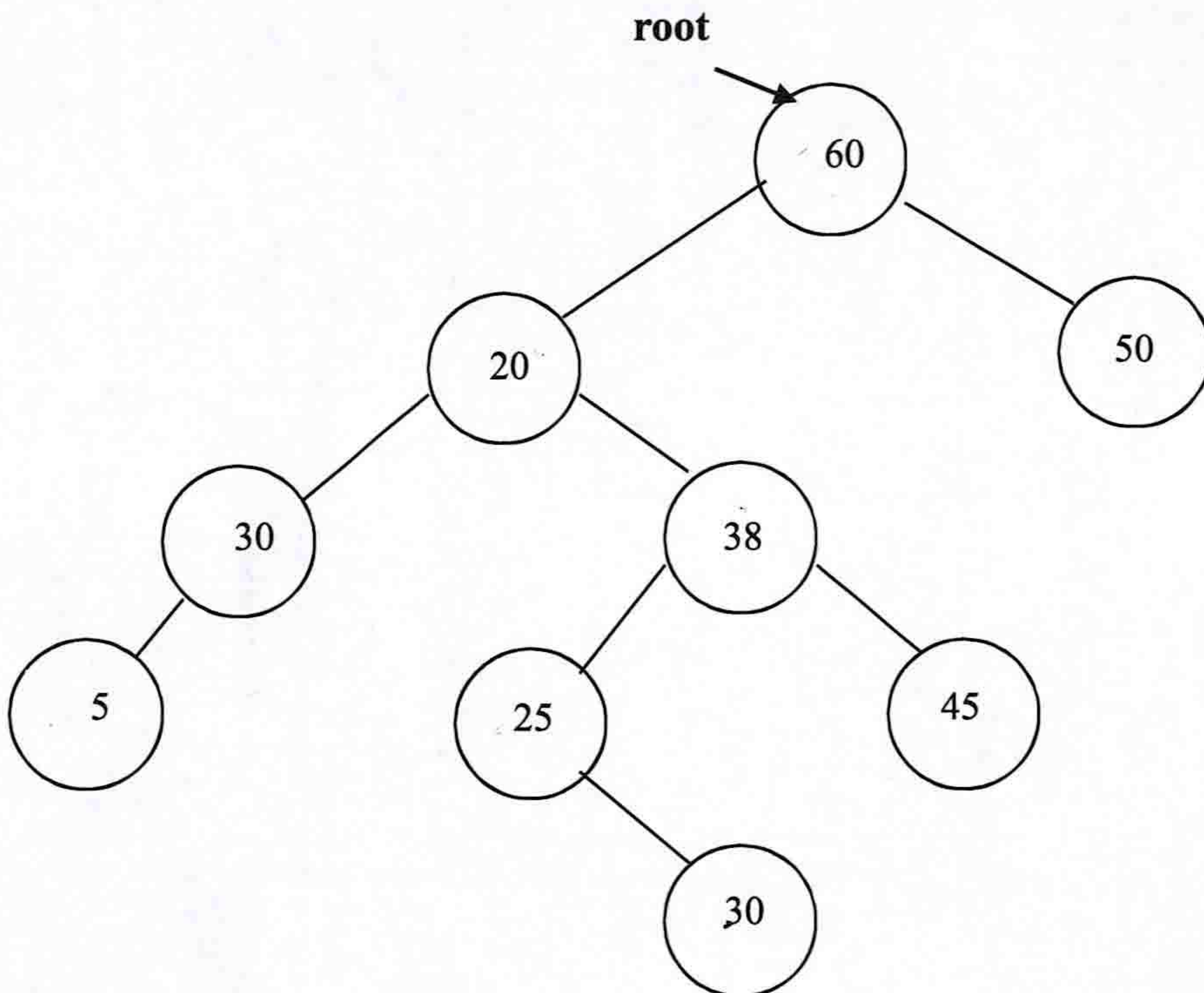


```

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

```

נתון העץ הבינרי הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

2 נק' (א) איזה ערך תחזיר השיטה **what** בעקבות הקריאה `BinaryTree.what(root)`?

התשובה היא:

9

3 נק' (ב) מה מבצעת השיטה **what** 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?

התשובה היא:

מחזיר את כמות העלים בעץ

2) (נקי) (ג) איזה ערך תחזיר השיטה something בעקבות הקריאה

BinaryTree.something (root, Integer.MIN_VALUE, Integer.MAX_VALUE)

התשובה היא:

False

3) (נקי) (ד) מה מבצעת השיטה something באופן כללי כשהיא מקבלת כפרמטרים שורש של עץ

בינרי root , min = Integer.MIN_VALUE , max = Integer.MAX_VALUE ? 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 משמעותו של הערך שהשיטה מחזירה. התייחסו למקרי קצה.

התשובה היא:

השיטה בודקת האם העץ הוא עץ חיפוש בינארי
אם העץ ריק הוא יחזיר אמת
אם העץ לא חיפוש בינארי הוא יחזיר שקר

3) (נקי) (ה) איזה ערך תחזיר השיטה secret בעקבות הקריאה BinaryTree.secret(root)?

התשובה היא:

4

4) (נקי) (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?

התשובה היא:

השיטה בודקת האם קיים עץ או תת עץ שהוא עץ חיפוש בינארי
ומחזירה את כמות העלים שקיימים בו

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

```
public class A
{
    private static int counterA = 0;
    private String _txt;
    public A (String txt)
    {
        counterA++;
        _txt = txt;
        System.out.println (A.counterA + ":" + txt);
    }

    public A ()
    {
        this ("Special");
    }

    public String getText()
    {
        return _txt;
    }

    public void setText (String txt)
    {
        _txt = txt;
    }

    public String toString ()
    {
        return getText();
    }
}

//-----
public class B extends A
{
    private int _times;
    public B (String txt, int times)
    {
        super (txt);
        System.out.println ( "times : " + times);
        _times = times;
    }

    public B ()
    {
        _times = 2;
    }
}
```



```

public int getTimes()
{
    return _times;
}

public String getText()
{
    String s = "";
    for (int i = 0; i < _times; i++)
        s+= super.getText();
    return s;
}
}
//-----

public class C extends B
{
    private int _lines;
    public C (String txt, int times, int lines)
    {
        super (txt,times);
        System.out.println (lines + " lines");
        _lines = lines;
    }

    public C (String txt, int t)
    {
        this (txt, t, t);
    }

    public String getText ()
    {
        String s = "";
        for (int i = 0; i < _lines; i++)
            s+= super.getText() + "\n";
        return s;
    }
}

```

בפרויקט נמצאת גם המחלקה Driver ובה השיטה main שלהלן:

```

public static void main (String[] args)
{
    - A[] a = new A[6];
    - a[0] = new A("Good Luck");
    - a[1] = new A();
    - a[2] = new B("OyVey", 2);
    - a[3] = new B( );
    a[4] = new C("Submarines", 2, 3);
    a[5] = new C("Planes",3);

    for (int i = 0; i < a.length; i ++)
        System.out.println (i + "***" + a[i]);
}

```


סעיף א (9 נקודות):

לאחר הרצת הקוד לעיל בשיטה main שבמחלקה Driver, המגדיר את האובייקטים שבמערך a, מלאו את הטבלה להלן כך שהיא תכיל בכל תא את הערך של התכונה המתאימה של האובייקט. אם לאובייקט אין תכונה כזו, סמנו ב- X

| האובייקט | _txt | _times | _lines |
|----------|------------|--------|--------|
| a[0] | Good Luck | X | X |
| a[1] | Special | X | X |
| a[2] | OyVey | 2 | X |
| a[3] | Special | 2 | X |
| a[4] | Submarines | 2 | 3 |
| a[5] | Planes | 3 | 3 |

סעיף ב (6 נקודות):

מה יודפס על הפלט לאחר הרצת הלולאה בקוד שבשיטה main לעיל? רק הלולאה, לא ההצהרות שלפניה.

התשובה היא (לא בהכרח כל השורות יתמלאו):

```
0 *** Good Luck
1 *** Special
2 *** OyVeyOyVey
3 *** SpecialSpecial
4 ** SubmarinesSubmarines
SubmarinesSubmarines
SubmarinesSubmarines
5 *** PlanesPlanesPlanes
PlanesPlanesPlanes
PlanesPlanesPlanes
```


שאלה 5 (18 נקודות)

נתונה המחלקה 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 h ) {
        _head = h;
    }

    // כאן יש עוד בנאים ושיטות...

    // המשך המחלקה בעמוד הבא
}
```



```

public IntNode what()
{
    IntNode j = _head;
    int p = 0;
    for (IntNode i = _head; i != null; i = i.getNext())
    {
        if (i.getValue() < p)
        {
            int temp = i.getValue();
            i.setValue(j.getValue());
            j.setValue(temp);

            j = j.getNext();
        }
    }
    return j;
}

public void secret()
{
    IntNode p = what();

    IntNode n = _head;
    while (p != null && n != p)
    {
        {
            int temp = n.getValue();
            n.setValue(p.getValue());
            p.setValue(temp);

            p = p.getNext();
            n = n.getNext().getNext();
        }
    }

    . . . // other methods
}

```

אתם יכולים להניח שהרשימה מלאה במספרים שלמים.

בטענות להלן, נסמן את איברי הרשימה כמספרים מופרדים בפסיקים, בתוך סוגריים מסולסלים.

כך לדוגמא, נסמן { -4, 9, 12, 21 } את הרשימה $-4 \rightarrow 9 \rightarrow 12 \rightarrow 21 \rightarrow \text{null}$

סעיף א (3 נקודות)

נתונה הרשימה $list = \{ -5, -4, -6, 8, 2, 9 \}$. מה יהיה הערך המספרי הנתון בחוליה החוזרת כתוצאה מהפעלת השיטה `list.what()`?

התשובה היא 8

סעיף ב (4 נקודות)

נתונה הרשימה $list = \{ -5, -4, -6, 8, 2, 9 \}$. איך תיראה הרשימה לאחר הקריאה לשיטה `list.secret()`?

התשובה היא 8, -4, 2, -5, 9, -6

סעיף ג (4 נקודות)

נתונה הרשימה $list = \{ -2, 2, 2, 3, 2, -2 \}$. איך תיראה הרשימה לאחר הקריאה לשיטה `list.secret()`?

התשובה היא 2, -2, 3, -2, 2, 2

סעיף ד (7 נקודות)

מה מבצעת השיטה `secret` 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. התייחסו למקרי קצה.

התשובה היא

השיטה משנה את סדר האיברים ברשימה לכך בראש הרשימה יהיה איבר חיובי לאחר מכן אחריו שלישי וכן הלאה מקרי קצה : אם כל איברי הרשימה חיובים או שליליים הרשימה לא תשתנה

בהצלחה !