

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

שאלה 1 - 25 נקודות

נתונות שתי מחרוזות תווים str1 ו-str2. כמו כן, קיימות שתי פעולות שאפשר לבצע על מחרוזת תווים:

1. להוסיף תו למחרוזת

2. למחוק תו מהמחרוזת

כדי להמיר (convert) מחרוזת אחת לשניה, אפשר להשתמש אך ורק בשתי הפעולות האלו. אנחנו מעוניינים להמיר מחרוזת אחת לשניה **במינימום** פעולות. **לדוגמא,**

- `str1 = "geek", str2 = "gesek"` : אפשר להמיר את str1 ל-str2 על ידי הוספת התו 's'
- `str1 = "sunday", str2 = "saturday"` : שימו לב שהאות הראשונה ושלוש האותיות האחרונות בשתי המחרוזות שוות, ולכן בעצם אנחנו צריכים להמיר את "un" ל-"atur". ההמרה הזו יכולה להתבצע בעזרת 4 פעולות בלבד: הוספת התו 'a', הוספת התו 't', מחיקת התו 'u' והוספת התו 'r'.

עליכם לכתוב שיטה **רקורסיבית** המקבלת כפרמטרים שתי מחרוזות str1 ו-str2 ומחזירה את מספר פעולות ההוספה והמחיקה המינימלי שצריך לבצע כדי להמיר את המחרוזת str1 למחרוזת str2.

שימו לב, אין צורך להחזיר את הפעולות עצמן ולא להדפיס אותן. רק להחזיר מהו המספר המינימלי של הפעולות הנדרשות.

חתימת השיטה היא:

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

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

אין צורך לדאוג ליעילות השיטה!

אל תשכחו לתעד את מה שכתבתם!

שאלה 2 - 25 נקודות

כתבו שיטה יעילה המקבלת כפרמטרים, מערך חד-ממדי arr המלא במספרים שלמים **הממוינים** **בסדר עולה ממש**, ומספר שלם num. השיטה מחזירה כמה שלשות של מספרים יש במערך arr שהסכום שלהם קטן ממש מ- num. המספרים בשלשות הם לא בהכרח מספרים עוקבים, ולא בהכרח נמצאים בתאים רצופים.

חתימת השיטה היא:

```
public static int countTriplets (int [] arr, int num)
```

לדוגמא,

- עבור המערך arr הבא, והערך num = 2

0	1	2	3
-2	0	1	3

יוחזר הערך 2 שכן אלו השלשות שהסכום של כל אחת מהן קטן ממש מ- 2:
(-2, 0, 1) (-2, 0, 3)

- עבור המערך arr הבא, והערך num = 12

0	1	2	3	4
1	3	4	5	7

יוחזר הערך 4 שכן אלו השלשות שהסכום של כל אחת מהן קטן ממש מ- 12:
(1, 4, 5) (1, 3, 7) (1, 3, 5) (1, 3, 4)
שימו לב שהסכום של השלשה (1, 4, 7) הוא בדיוק 12 ולכן היא לא נספרת.

אפשר להניח שהמערך מלא במספרים שלמים שונים זה מזה והוא ממוין בסדר עולה ממש. 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. רמז: הסיבוכיות הנדרשת כאן היא ריבועית.
- כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.
- אל תשכחו לתעד את מה שכתבתם!

**חלק ב - את התשובות לשאלות 3-6 יש לכתוב על גבי השאלון.
לא נבדוק תשובות שייכתבו במקום אחר!**

שאלה 3 - 12 נקודות

א. נניח שהמחלקה Node שלהלן מממשת צומת של עץ בינרי.

```
public class Node
{
    private int _number;
    private Node _leftSon, _rightSon;

    public Node (int num)
    {
        _number = num;
        _leftSon = null;
        _rightSon = null;
    }

    public int  getNumber()      {return _number; }
    public Node getLeftSon()     {return _leftSon; }
    public Node getRightSon()    {return _rightSon; }

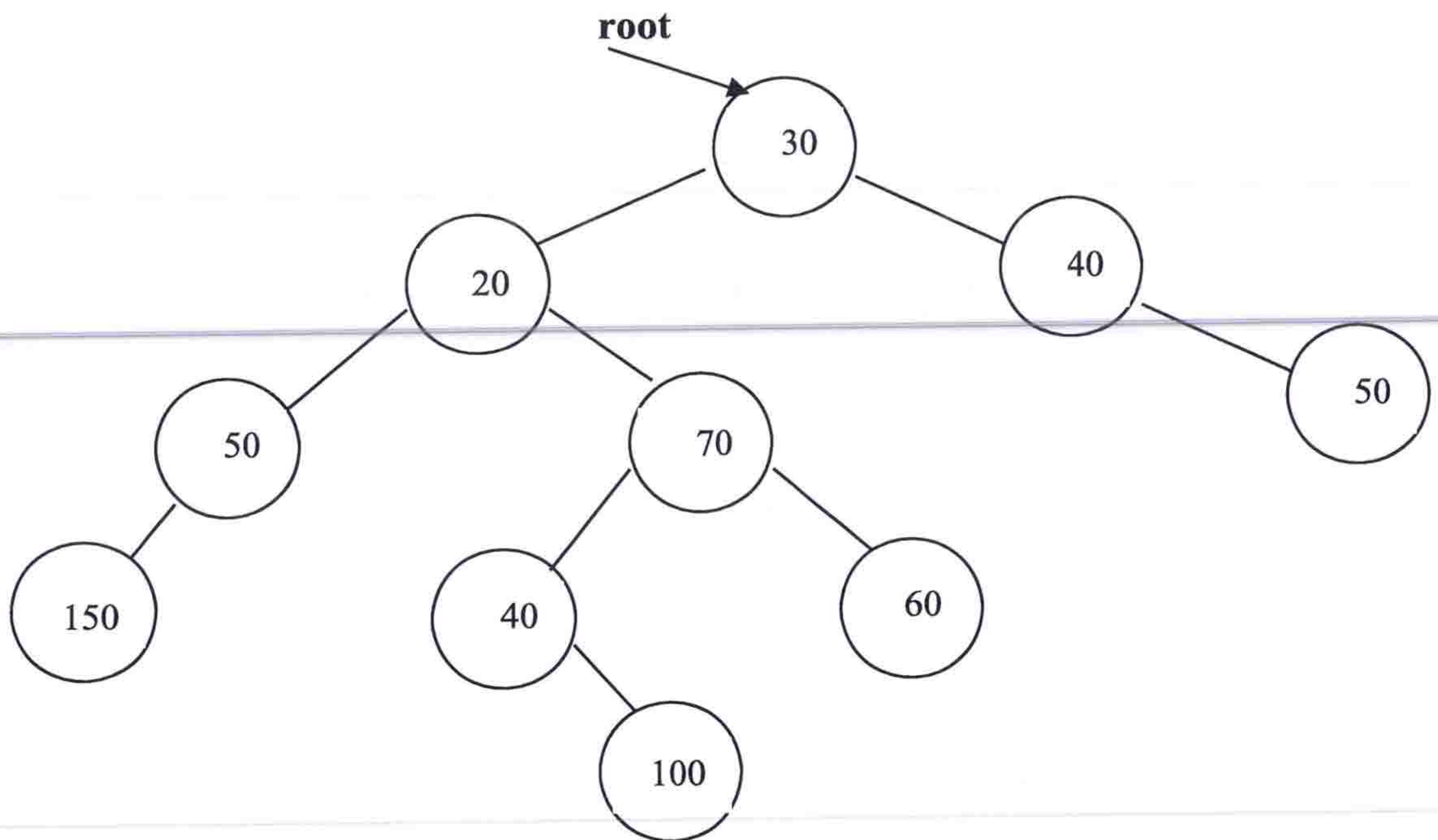
    public void setNumber(int number) { _number = number; }
    public void setLeftSon(Node node) { _leftSon = node; }
    public void setRightSon(Node node){ _rightSon = node; }
}
```

נתונה השיטה what הבאה שמקבלת כפרמטר את node מטיפוס Node שמצביע לשורש של עץ בינרי.

```
public static int what(Node node)
{
    int x=0;
    int y=0;
    if (node.getLeftSon()!=null)
        x = what (node.getLeftSon());

    if (node.getRightSon()!=null)
        y = what (node.getRightSon());

    if (x > y)
        return (node.getNumber() + x);
    else
        return (node.getNumber() + y);
}
```

ענו על שני הסעיפים הבאים:

5 נק') (i) איזה ערך תחזיר השיטה what אם נקרא לה עם השורש root שלעיל ?
התשובה היא:

7 נק') (ii) מה מבצעת השיטה what באופן כללי? הסבירו בקצרה מה עושה השיטה 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 הקצה השונים.

התשובה היא:

שאלה 4 - 12 נקודות

נתונות המחלקות הבאות (כל מחלקה בקובץ נפרד, כמובן):

```
public class A {
    private int x;
    public A() {
        this.x = 3;
    }
    public A(int x) {
        this.x = x;
        System.out.println(toString());
    }
    public int getX() {
        return x;
    }
    public void f() {
        if(x % 2 == 0)
            System.out.println("Even");
        System.out.println("In A's f");
        g();
    }
    public void g() {
        System.out.println("In A's g");
    }
    public String toString() {
        return "X: " + this.x;
    }
}
```

```
public class B extends A{
    private int y;
    public B() {
        super(4);
        y = getX() + 2;
    }
    public B(int y) {
        super();
        this.y = y;
        System.out.println(toString());
    }
    public void h() {
        System.out.println("in B's h");
    }
    public void f() {
        System.out.println("in B's f");
        g();
        h();
    }
    public String toString() {
        return super.toString() + " Y: " + y;
    }
}
```



```
public class Main {
    public static void main(String[] args) {
        // ****
    }
}
```

בכל אחד מהסעיפים הבאים נתון קוד. עבור כל סעיף כתבו מה יקרה אם נכתוב את הקוד בתוך ה-main במקום השורה המסומנת ב **** // (כל סעיף עצמאי והסעיפים אינם תלויים אחד בשני). כתבו האם תהיה שגיאת קומפילציה (כתבו מדוע) או שתהיה שגיאת ריצה (כתבו מדוע) או שכתבו מה יהיה הפלט על המסך. 2 נקודות לכל סעיף. מספר השורות הריקות לא בהכרח מתאים לפלט.

א.

```
A a1 = new A();
a1.f();
```

ב.

```
A a2 = new B(5);
a2.h();
```

ג.

```
A a3 = new B(2);  
B b1 = (B) a3;  
b1.f();
```

ד.

```
A a4 = new B();  
a4.f();  
System.out.println(a4.toString());
```

ה.

```
B b2 = (B) (new A(6));  
System.out.println(b2.toString());
```



```
B b3 = (A) (new B());
System.out.println(b3.toString());
```

שאלה 5 - 16 נקודות

לפניכם המחלקות IntNode ו-IntList. כל אחת בקובץ נפרד, כמובן.

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

    public IntNode (int data, IntNode node)
    {
        _data = data;
        _next = node;
    }
    public IntNode getNext()
    {
        return _next;
    }
    public void setNext(IntNode node)
    {
        _next = node;
    }
    public int getData()
    {
        return _data;
    }
    public void setData (int data)
    {
        _data = data;
    }
}

//-----
```

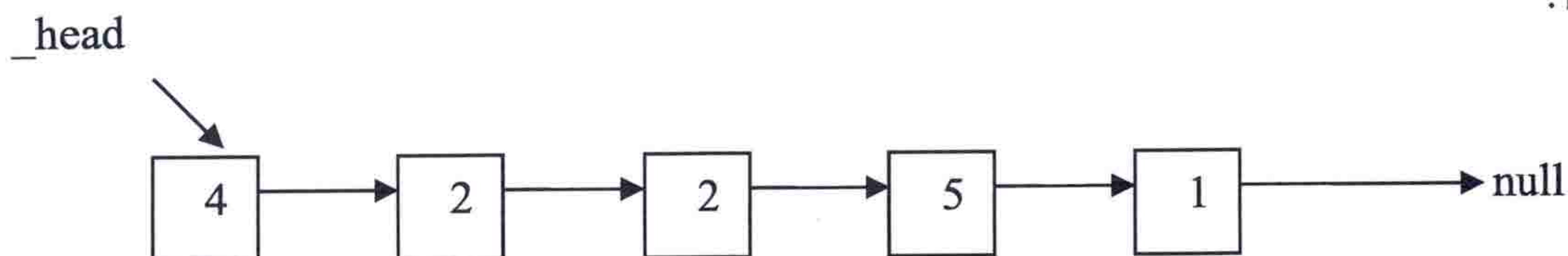


```

public class IntList
{
    private IntNode _head;
    public IntList()
    {
        _head = null;
    }
}

```

נתונה רשימה מקושרת של קודקודים מסוג IntNode, כאשר _head מצביע על האיבר הראשון ברשימה:



אנחנו נכתוב אותה כך {4, 2, 2, 5, 1}

השיטה הבאה (method1) נמצאת בתוך המחלקה IntList:

```

public int method1()
{
    int m1 = 0, m2 = 0;

    for(IntNode p = _head; p != null; p = p.getNext())
    {
        m2 = m2 + p.getData();
        if (m2 < 0)
            m2 = 0;
        if (m1 < m2)
            m1 = m2;
    }
    return m1;
}

```

סעיף א: (3 נקודות)

נפעיל את השיטה method1 על הרשימה {1, 5, 4, 2}. איזה ערך תחזיר השיטה?

✓

סעיף ב: (5 נקודות)

נפעיל את השיטה method1 על הרשימה {-2, -3, 4, -2, 1, 5, -3}. איזה ערך תחזיר השיטה?

✓

סעיף ג: (8 נקודות)

מה מבצעת השיטה method1 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. מה משמעותו של הערך המוחזר מהשיטה method1? התייחסו למקרי קצה.

שאלה 6-10 נקודות

השיטה הבאה אמורה למצוא איבר x בתוך מערך חד-ממדי a בעזרת חיפוש בינרי. המערך מלא במספרים שלמים מסודרים בסדר לא-יורד.

```
public static void f(int[] a, int x)
{
    int i, j, k;
    i = 0; j = a.length-1;
    do {
        k = (i + j) / 2;
        if( a[k] < x)
            i = k;
        else
            j = k;
    } while(a[k] != x && i < j);
    if(a[k] == x)
        System.out.println ("x is in the array " );
    else
        System.out.println(" x is not in the array " ) ;
}
```


סעיף א (5 נקודות)

1. נתונות שורות הקוד הבאות בשיטה הראשית main :

```
int[] a1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int x1 = 9;
```

מה יקרה כתוצאה מהקריאה לשיטה f(a1, x1) ? הסבר את תשובתך

—
—

2. נתונות שורות הקוד הבאות:

```
int[] a2 = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
int x2 = 0;
```

מה יקרה כתוצאה מהקריאה לשיטה f(a2, x2) ? הסבר את תשובתך

—
—
—

3. נתונות שורות הקוד הבאות:

```
int[] a3 = {2, 2, 2, 2, 2, 2, 2, 2, 2, 2};
int x3 = 3;
```

מה יקרה כתוצאה מהקריאה לשיטה f(a3, x3) ? הסבר את תשובתך

—
—

4. נתונות שורות הקוד הבאות:

```
int[] a4 = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
int x4 = 16;
```

מה יקרה כתוצאה מהקריאה לשיטה f(a4, x4) ? הסבר את תשובתך

5. נתונות שורות הקוד הבאות:

```
int[] a5 = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};  
int x5 = 20;
```

מה יקרה כתוצאה מהקריאה לשיטה $f(a5, x5)$? הסבר תשובתך

סעיף ב (5 נקודות)

איזה שינוי (אם בכלל) צריך לעשות כדי לתקן את השיטה f שלעיל? סמנו תשובתכם על השאלון.

1. לשנות את תנאי ה- if הראשון לתנאי הזה:

```
if (a[k] < x) i = k-1; else j = k+1;
```

2. לשנות את תנאי ה- if הראשון לתנאי הזה:

```
if (a[k] < x) i = k+1; else j = k-1;
```

3. לשנות את תנאי ה- if הראשון לתנאי הזה:

```
if (a[k] <= x) i = k; else j = k;
```

4. לשנות את תנאי ה- $while$ לתנאי הזה:

```
while ((a[k] == x) && (i < j));
```

5. אין צורך לשנות כלום. הקוד תקין.

בהצלחה