



מס' שאלון - 460

9

במרץ 2015

מסטר 2015א

20441 / 4

מס' מועד 89

שאלון בחינת גמר

20441 - מבוא למדעי המחשב ושפת Java

משך בחינה: 3 שעות

בשאלון זה 11 עמודים

מבנה הבחינה:

- * קראו בעיון את ההנחיות שלהלן:
- * בבחינה יש שש 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 - 6 יש לכתוב על גבי השאלון. לא נבדוק תשובות שייכתבו במקום אחר!
- * אפשר לתעד בעברית. אין צורך בתיעוד API.

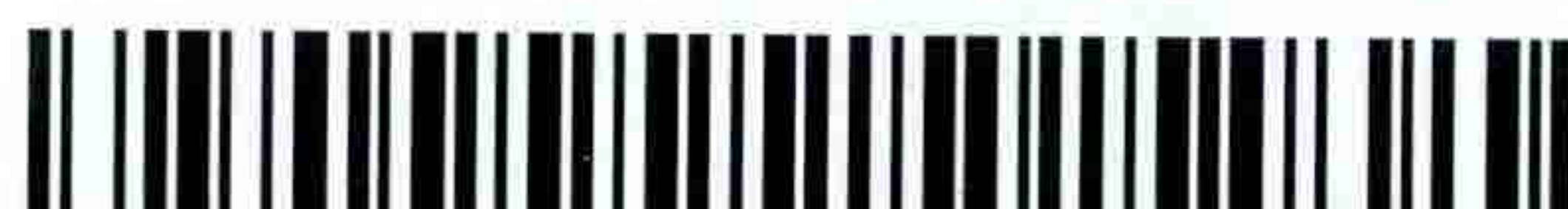
חומר עזר:

ספר הלימוד : java software solutions מאת : lewis/loftus
חוברת השקפים של הקורס.
מותרות הערות בכתב יד, ע"ג הספרים.
אין להכניס חומר מודפס או כל חומר אחר מכל סוג שהוא.

בהצלחה !!!

החזירו

למשגיח את השאלון
וכל עזר אחר שקיבלתם בתוך מחברת התשובות



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

שאלה 1 - 25 נקודות

כתבו שיטה רקורסיבית בוליאנית המקבלת מערך מלא במספרים שלמים, ומחזירה true אם אפשר לחלק את איברי המערך לשתי קבוצות, לאו דווקא שוות בגודלן, ולא דווקא בתאים סמוכים, כך שמכפלת איברי קבוצה אחת שווה למכפלת איברי הקבוצה השנייה. אם אי אפשר, השיטה תחזיר false.
חתימת השיטה היא:

```
public static boolean splitEqualMult(int [] a)
```

דוגמאות:

- עבור המערך:

| | | | | | |
|---|----|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 15 | 3 | 4 | 2 | 5 |

התשובה שתוחזר תהיה true שכן למשל, $2 * 3 * 2 * 5 = 15 * 4$.

- עבור המערך הבא:

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 4 | 6 | 2 | 3 | 4 |

התשובה שתוחזר תהיה false שכן אין אף חלוקה של איברי המערך לקבוצות כך שמכפלת איברי קבוצה אחת שווה למכפלת איברי הקבוצה השנייה.

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

אפשר להניח כי המערך אינו ריק.

אפשר להשתמש בהעמסת-יתר (overloading).

אל תשכחו לתעד את מה שכתבתם!

שאלה 2 - 25 נקודות

מערך נקרא "ממוין לסרוגין" אם שני התת-מערכים שלו המורכבים אחד ממספרים המופיעים באינדקסים הזוגיים (0, 2, ...) והשני ממספרים המופיעים באינדקסים האי-זוגיים (1, 3, ...) הם ממוינים כל אחד. לדוגמא, המערך {1, 2, 5, 3, 6, 10, 9} הוא ממוין לסרוגין, בעוד שהמערך {1, 2, 5, 3, 4, 10, 9} הוא לא (כי 4 מופיע אחרי 5 בתת-מערך הזוגיים).

כתבו שיטה `findX` המקבלת מערך ממוין לסרוגין של שלמים `a` ומספר שלם `x` ומחזירה `true` אם קיימים ב-`a` שני איברים סמוכים שסכומם הוא `x`, ו-`false` אחרת. לדוגמא, עבור המערך {1, 2, 5, 3, 6, 10, 9} ו-`x=9`, השיטה תחזיר `true` (שני האיברים הסמוכים 3 ו-6 סכומם 9) ועבור אותו מערך `a` כאשר `x=5` השיטה תחזיר `false`. (אין שני איברים סמוכים שסכומם 5). יש להתייחס גם למקרה בו במערך יש רק איבר אחד, ואז השיטה צריכה להחזיר `false`.

השיטה צריכה להיות יעילה ככל האפשר מבחינת זמן ומבחינת מקום. שיטה שאינה יעילה מספיק כלומר, שתהיה בסיבוכיות גדולה יותר מזו הנדרשת לפתרון הבעיה תקבל ניקוד חלקי בלבד.

אל תשכחו לתעד את מה שכתבתם!

כתבו מה סיבוכיות הזמן וסיבוכיות המקום של השיטה שכתבתם.

שימו לב: אסור להשתמש במחלקות מוכנות כבר של `Java`. מותר להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים, ובלבד שתהיה הפניה מדויקת אליהן.

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

שאלה 3 - 16 נקודות

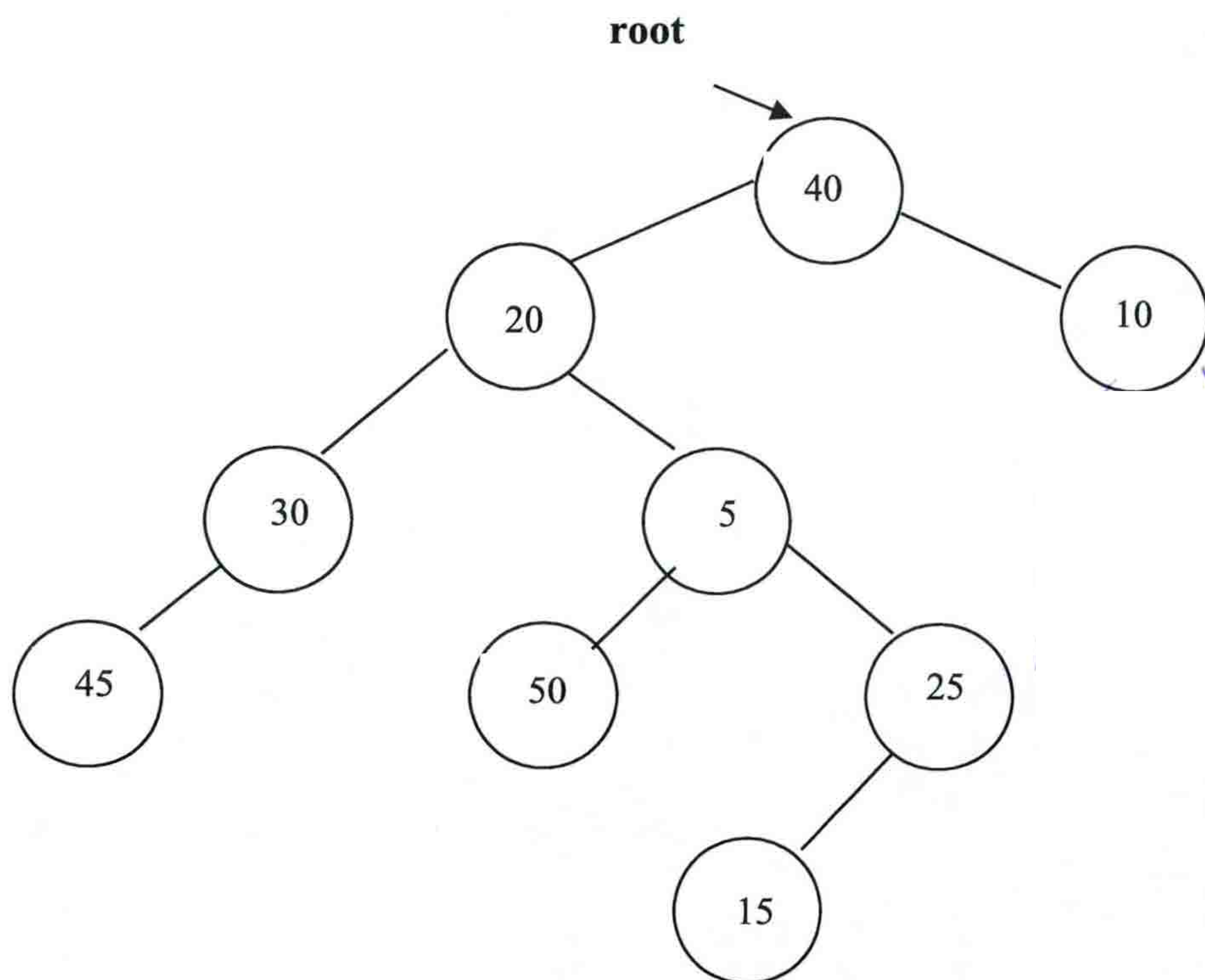
נניח שהמחלקה Node שלהלן מממשת עץ בינרי.

```
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; }
}
```

נתון העץ הבינרי הבא, ששורשו הוא root



להלן נתונות שיטות סטטיות שונות, המבצעות פעולות שונות על עצים בינריים.
השיטה max מקבלת כפרמטרים שני מספרים שלמים a ו-b ומחזירה את הגדול ביניהם. הניחו
שהמימוש שלה כבר כתוב.

```
public static int max (int a, int b) {...}

public static int f (Node t)
{
    if (t == null)
        return 0;
    return 1 + max (f (t.getLeftSon()), f (t.getRightSon()));
}

public static boolean what (Node root)
{
    if (root == null)
        return true;
    int num = f(root);
    return what(root, 1, num);
}

private static boolean what (Node root, int x, int num)
{
    if ((root.getLeftSon()==null) && (root.getRightSon()==null))
    {
        if (x == num)
            return true;
        else
            return false;
    }
    if (((root.getLeftSon()==null)&& (root.getRightSon()!=null))
        ||
        ((root.getLeftSon()!=null)&& (root.getRightSon()==null)))
    return false;
    return what (root.getLeftSon(), x+1, num)
        && what (root.getRightSon(), x+1, num);
}
```


סעיף א. (3%)

מה מבצעת השיטה f בהינתן לה כפרמטר t מטיפוס Node שמצביע לשורש של עץ בינרי? כתבו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת.

סעיף ב. (3%)

איזה ערך תחזיר השיטה **what** בהינתן לה כפרמטר root מטיפוס Node שמצביע לשורש של העץ בינרי המצויר לעל?

סעיף ג. (4%)

מה צריך לשנות בעץ המצויר לעיל כדי שהשיטה **what** תחזיר ערך אחר מהערך שהיא מחזירה (שכתבתם בסעיף ב)? כתבו במפורש ובמפורט אילו צמתים צריך להוריד או להוסיף או לשנות בעץ. כתבו מספר מינימלי של שינויים, כדי שהתשובה תהיה שונה.
(מספר השורות הנתונות כאן הוא אקראי ולא קשור למספר השינויים הדרושים).

סעיף ד. (6%)

מה מבצעת השיטה **what** בהינתן לה כפרמטר t מטיפוס Node שמצביע לשורש של עץ בינרי? כתבו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת.

שאלה 4 - 12 נקודות

נתונה המחלקה הבאה:

```
public abstract class A
{
    private int _num;

    public A() {
        _num = 2;
    }


    public int getNum() {
        return _num;
    }

    public abstract boolean f(int x);
}
```

בכל אחד מהסעיפים הבאים נתונה הגדרה של מחלקה B היורשת מ-A. בכל סעיף כתבו האם קוד המחלקה תקין, או אם לא עוברת קומפילציה הסבירו למה. כל סעיף שווה 2 נקודות.

1.


```
public class B extends A {
    public boolean f(double x)
    { return x == 2.0; }
}
```



-
-

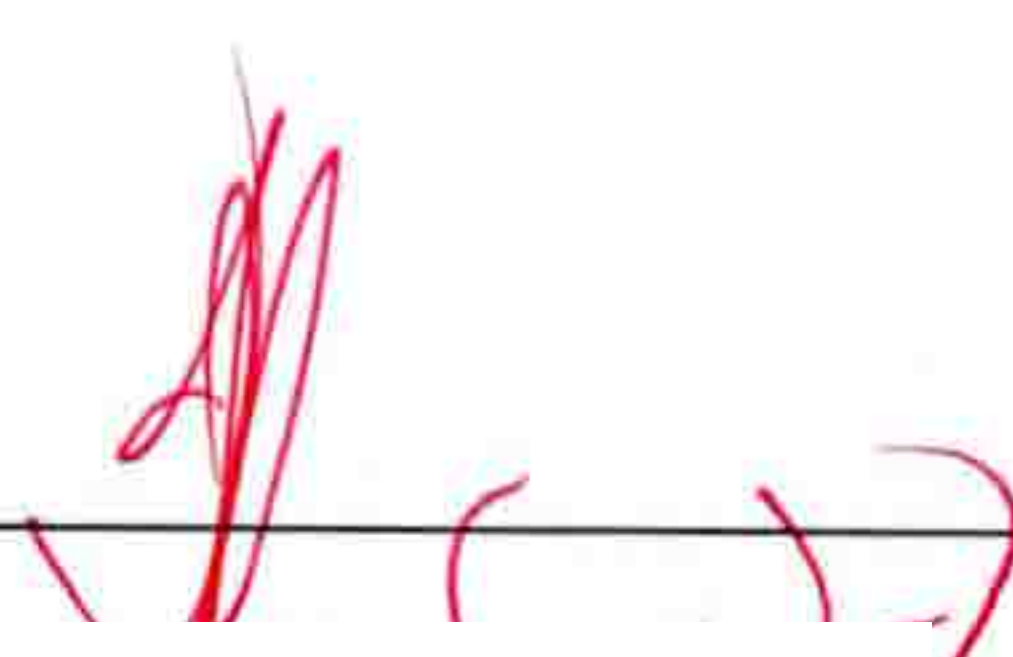
2.

```
public abstract class B extends A {
    public boolean f(int x)
    { return x == 2; }
}
```



3.

```
public class B extends A {
    public boolean f(int x)
    { return x == _num; }
}
```



.4

```
public abstract class B extends A {  
    public B(int n) {  
        super(n);  
    }  
}
```

.5

```
public abstract class B extends A {  
    public B(int n) {  
        n += getNum();  
        System.out.println(n);  
    }  
}
```

.6

```
public class B extends A {  
    public boolean f(int x) {  
        Object obj1 = new A();  
        (A)obj1.f(x);  
    }  
}
```

שאלה 5 - 15 נקודות

נתונה המחלקה `DataQueue` הבאה, המממשת תור. איברי התור הם אובייקטים מהמחלקה `TreeNode`, המכילים ערכים מטיפוס `int`:

```
public class DataQueue
{
    private TreeNode _rear, _front;

    public DataQueue() {...}
    // Constructs an empty queue

    public boolean empty() {...}
    // Returns true iff the queue is empty

    public void enqueue(TreeNode newItem) {...}
    // Adds the new item to the end of the queue

    public TreeNode retrieve(){...}
    // Removes oldest item from the queue and returns it.
    // Returns null if queue is empty.
}
```

בנוסף, נתונות שתי השיטות הסטטיות הבאות:

```
public static boolean mystery (DataQueue q1, DataQueue q2)
{
    if (q1.empty() && q2.empty())
        return true;
    if (q1.empty() || q2.empty())
        return false;
    if (!(q1.retrieve().equals( q2.retrieve())))
        return false;
    return mystery (q1, q2);
}

public static void secret (DataQueue q)
{
    if (!q.empty())
    {
        TreeNode node = q.retrieve();
        secret (q);
        q.enqueue (node);
    }
}
```

השיטה `equals` המופיעה בשיטה `mystery` שייכת למחלקה `TreeNode` והיא מחזירה `true` אם האובייקט שעליו מופעלת השיטה שווה לאובייקט המועבר לה כפרמטר בשדה `data` (מטיפוס `int`).

סעיף א (5 נקודות)

מה מבצעת השיטה `mystery` כאשר היא מקבלת כפרמטרים שני תורים `q1` ו-`q2`?
כתבו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת.

סעיף ב (5 נקודות)

מה מבצעת השיטה `secret` כאשר היא מקבלת כפרמטר תור `q`?
כתבו בקצרה מה מבצעת השיטה ולא כיצד היא מבצעת זאת.

סעיף ג (5 נקודות)

תנו דוגמא לשני תורים `q1` ו-`q2`, כך שבכל תור ישנם בין 3 ל-5 איברים שונים מטיפוס `int`, עבורם
רצף הפקודות הבא ידפיס את הערך `true`.

```
secret (q2);  
System.out.println (mystery(q1, q2));
```



שאלה 6 - 7 נקודות

נתונה השיטות הבאות:

```
public static void what(int n, int k)
{
    what(n, k, "");
}

private static void what(int n, int k, String s)
{
    if (k == 0)
        System.out.println(s);
    else if (n > 0)
    {
        what(n-1, k, s);
        what(n-1, k-1, n+s);
    }
}
```

א. מה תדפיס השיטה הציבורית what אם תתבצע הקריאה הבאה:

what (5, 3);

כתבו את תשובתכם כאן (מספר השורות המצוירות לא תואם בהכרח את מספר השורות בפלט): (4 נקודות)

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

ב. מה מבצעת השיטה what באופן כללי, בהינתן לה ערכים שלמים n ו-k? כתבו מה עושה השיטה ולא כיצד היא עושה זאת. שימו לב למקרי קצה. (3 נקודות)

-
-

בהצלחה!