

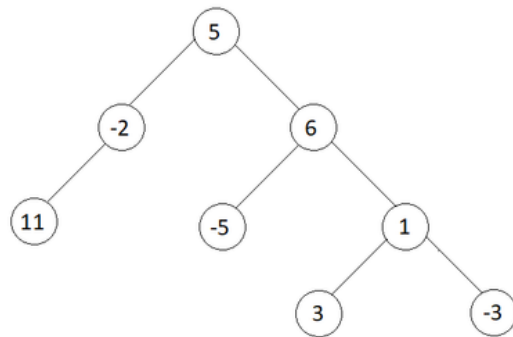
כללי – יש לכתוב קוד במחברת ולקבל אישור למימוש
בכל השאלות בהן יש התייחסות למחלקה Node, מדובר במחלקה Node שקיבלתם.

שאלה ראשונה

בקובץ Tester.java ממומשת סריקת inorder רקורסיבית.
יש לממש סריקת inorder בשימוש במחסנית (השתמשו במחסנית שאתם מימשתם).

שאלה שניה

כתבו שיטה המחזירה את מס' הרמה בעלת סכום צמתים מקסימלי.
בדוגמא הבאה ניתן לראות שרמה 2 היא בעלת סכום צמתים מקסי' ($11 - 5 + 1 = 7$)



חתימת השגרה:

```
static int maxLevelSum(Node root)
```

רמז: ניתן להשתמש בתור (שאתם מימשתם).

שאלה שלישית

נתונה השגרות הבאות:

```
public boolean g(Node t, int a, int b)
{
    if(t == null)
        return true;
    return t.getNumber() > a && t.getNumber() < b &&
        g(t.getLeftSon(), a, t.getNumber()) &&
        g(t.getRightSon(), t.getNumber(), b);
}
```

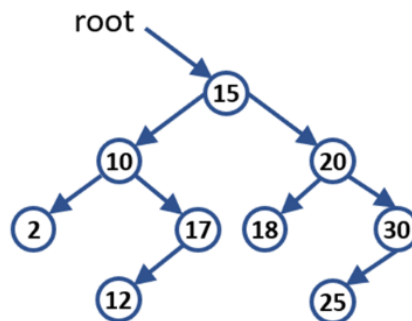
```

public static boolean f(Node t){
    if (t == null)
        return true;
    if (t.getLeftSon() == null && t.getRightSon() == null)
        return true;
    if (t.getRightSon() == null)
        return (t.getNumber() > t.getLeftSon().getNumber())
            && f(t.getLeftSon());
    if (t.getLeftSon() == null)
        return (t.getNumber() <= t.getRightSon().getNumber())
            && f(t.getRightSon());
    return (t.getNumber() <= t.getRightSon().getNumber())
        && (t.getNumber() > t.getLeftSon().getNumber())
        && f(t.getLeftSon())
        && f(t.getRightSon());
}

```

השיטה g מקבלת כפרמטר את t מטיפוס Node שמצביע לשורש של עץ בינרי, וכן את הערכים
 $a = \text{Integer.MIN_VALUE}$ (כלומר המספר השלם הקטן ביותר האפשרי),
 $b = \text{Integer.MAX_VALUE}$ (כלומר המספר השלם הגדול ביותר האפשרי).

א. בעבור העץ הבינרי הבא t, סמנו מה יחזירו השיטות f(t) ו-g(t):



השיטה f(t) תחזיר את הערך:

true ☐

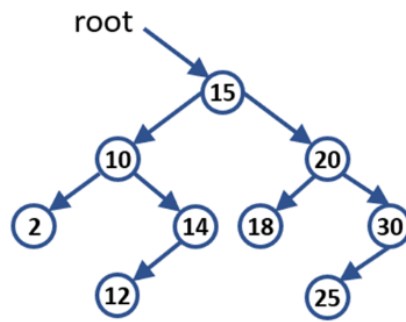
false ☐

השיטה g(t) תחזיר את הערך:

true ☐

false ☐

ב. בעבור העץ הבינרי הבא t , סמנו מה יחזירו השיטות $f(t)$ ו- $g(t)$:



השיטה $f(t)$ תחזיר את הערך:

true ☐

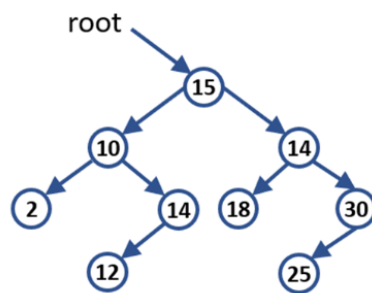
false ☐

השיטה $g(t)$ תחזיר את הערך:

true ☐

false ☐

ג. בעבור העץ הבינרי הבא t , סמנו מה יחזירו השיטות $f(t)$ ו- $g(t)$:



השיטה $f(t)$ תחזיר את הערך:

true ☐

false ☐

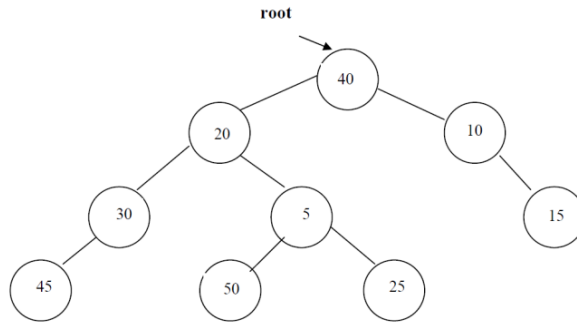
השיטה $g(t)$ תחזיר את הערך:

true ☐

false ☐

שאלה רביעית

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 int what (Node t) {
    return what(t, f(t));
}

private static int what (Node t, int num) {
    if (t == null)
        return 0;
    if (num == 1)
        return 1;
    return what(t.getLeftSon(), num- 1)
        + what(t.getRightSon(), num- 1);
}
```

א

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.

ב

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

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 חמישית:

כתבו בקצרה מה מבצעת השיטה foo {שקוראת ל-boo ול-what (לעיל שאלה 4 שיטה שלישית)}

```
public int foo(Node node)
{
    int maxWidth = 0;
    int width;
    int h = boo(node);
    int i;
    for (i = 1; i <= h; i++) {
        width = what(node, i);
        if (width > maxWidth)
            maxWidth = width;
    }
    return maxWidth;
}

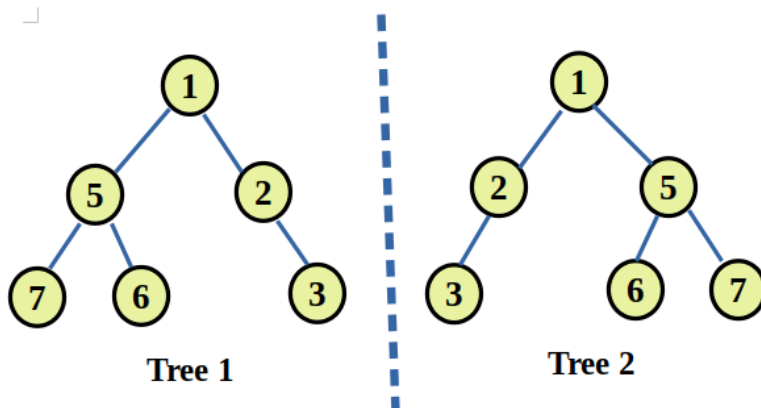
Private int boo(Node node)
{
    if (node == null)
        return 0;
    else {
        int lH = boo (node.left);
        int rH = boo (node.right);

        return (lH > rH)? (lH + 1): (rH + 1);
    }
}
```

שאלה שישית:

ממשו שיטה mirrorTree(Node root) המקבלת שורש של עץ, והפכת אותו לתמונת-ראי באופן רקורסיבי.

דוג':



חתימת הפונקציה היא public static Node mirrorTree(Node root)

(ניתן לממש בשני אופנים, ע"י יצירת עץ חדש, או רק להחליף את המפתחות).