נניח שהמחלקה Node שלהלן מממשת עץ חיפוש בינארי:

```
public class Node
{
        private int number;
        private Node leftSon;
        private Node rightSon;


        public int getNumber() {
                return number;
        }
        public Node getLeftSon() {
                return leftSon;
        }
        public Node getRightSon() {
                return rightSon;
        }
}
```

שאלה 1 – סריקת עץ בינארי

1. כזכור בהינתן עץ בינארי, **סריקה בסדר תחילי** (preorder traversal) מבקרת בשורש העץ ואחר כך פונה לקרוא את תתי העץ שלו, תחילה השמאלי ואחר כך הימני.

   ממשו שיטה סטטית המבצעת סריקה בסדר תחילי (preorder traversal) ומדפיסה את איבריו
   חתימת השיטה הינה: public static void preorder(Node root)

2. כזכור בהינתן עץ בינארי, **סריקה בסדר תוכי** (inorder traversal) קוראת את תת-העץ השמאלי, מבקרת בשורש, ולבסוף תת-העץ הימני.

   ממשו שיטה סטטית המבצעת סריקה בסדר תוכי (inorder traversal) ומדפיסה את איבריו
   חתימת השיטה הינה: public static void inorder(Node root)

3. כזכור בהינתן עץ בינארי, **סריקה בסדר סופי** (postorder traversal) קוראים את שני תתי העצים ורק אז עורכים ביקור בשורש.

   ממשו שיטה סטטית המבצעת סריקה בסדר סופי (postorder traversal) ומדפיסה את איבריו
   חתימת השיטה הינה: public static void postorder(Node root)

4. כזכור בהינתן עץ בינארי, **סריקת לפי רמות (level-order)** קוראים רמה אחרי רמה משמאל לימין אחל מרמה 0
ממשו שיטה סטטית המבצעת סריקה לפי רמות (levelOrder traversal) ומדפיסה את איבריו
חתימת השיטה הינה: public static void levelOrder (Node root)

שאלה 1

נניח שהמחלקה Node שלהלן מממשת 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; }
}
```

המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי.

בין השיטות נתונות השיטות הבאות: **שימו לב, השיטות מתייחסות לעץ חיפוש בינרי.**

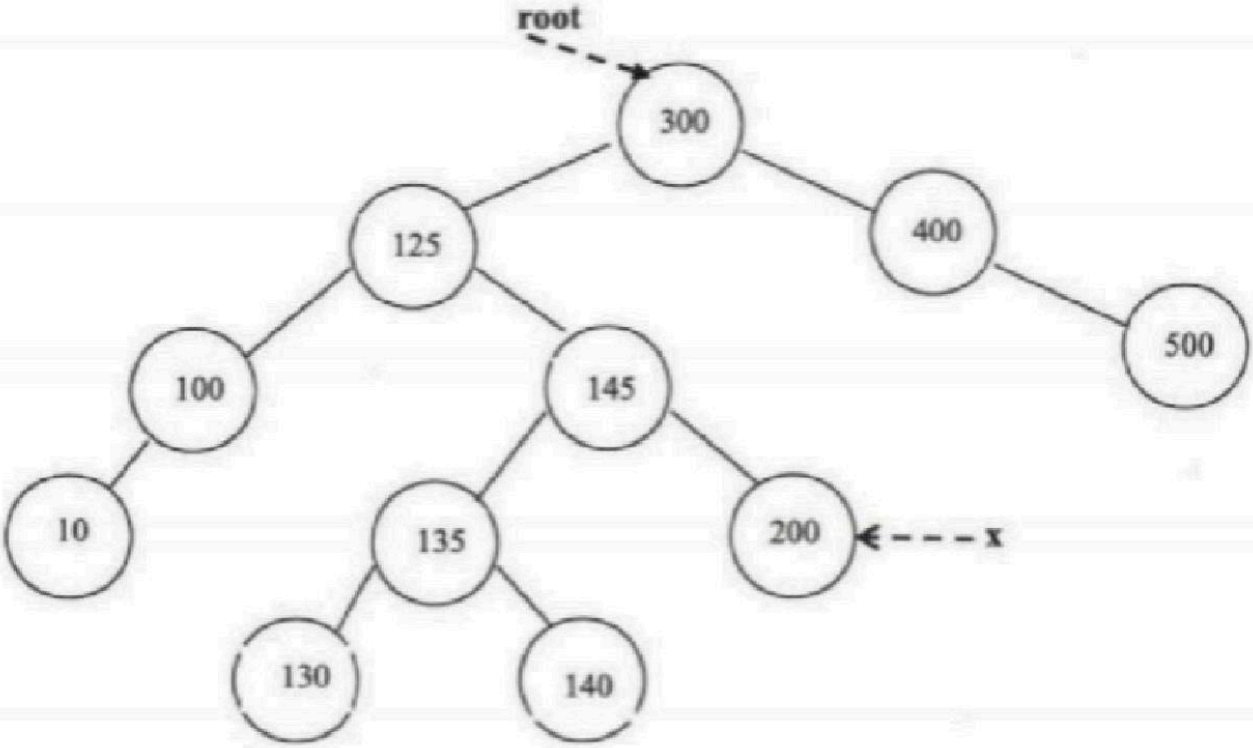
```
    public static Node what (Node root)
    {
        if (root==null)
            return null;
        if (root.getLeftSon() == null)
            return root;
        return what (root.getLeftSon());
    }

    public static Node secret (Node root, Node x)
    {
        if (x.getRightSon() != null)
            return what (x.getRightSon());
        Node temp = null;
        return secret (root, x, temp);
    }
```

שאלה 1 המשך

```
private static Node secret (Node root, Node x, Node p)
{
    if (root == null)
        return p;
    if (x.getNumber() < root.getNumber())
    {
        p = root;
        root = root.getLeftSon();
    }
    else if (x.getNumber() > root.getNumber())
        root = root.getRightSon();
    else
        return p;
    return secret (root, x, p);
}
```

נתון **עץ חיפוש בינרי** הבא, ששורשו הוא root



ענו על הסעיפים הבאים:

(i) איזה ערך תחזיר הקריאה `inaryTree.what(root)`

התשובה היא:

שאלה 1 המשך

(ii) מה מבצעת השיטה **what** באופן כללי? 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 המוחזר מהשיטה what.

התשובה היא:

(iii) איזה ע"ך תדפיס הפקודה (root ו- x מסמנים צמתים בעץ לעיל).

```
System.out.println ((BinaryTree.secret(root,x)).getNumber());
```

התשובה היא:

(iv) מה מבצעת השיטה **secret** באופן כללי בהינתן לה שורש של **עץ חיפוש בינרי** וצומת x כלשהו בעץ? 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, הסבירו מה המשמעות של הערך המוחזר מהשיטה secret. **התייחסו למקרי הקצה האפשריים.**

התשובה היא:

שאלה 2

המחלקה Node שלהלן מממשת צומת בעץ 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; }
}
```

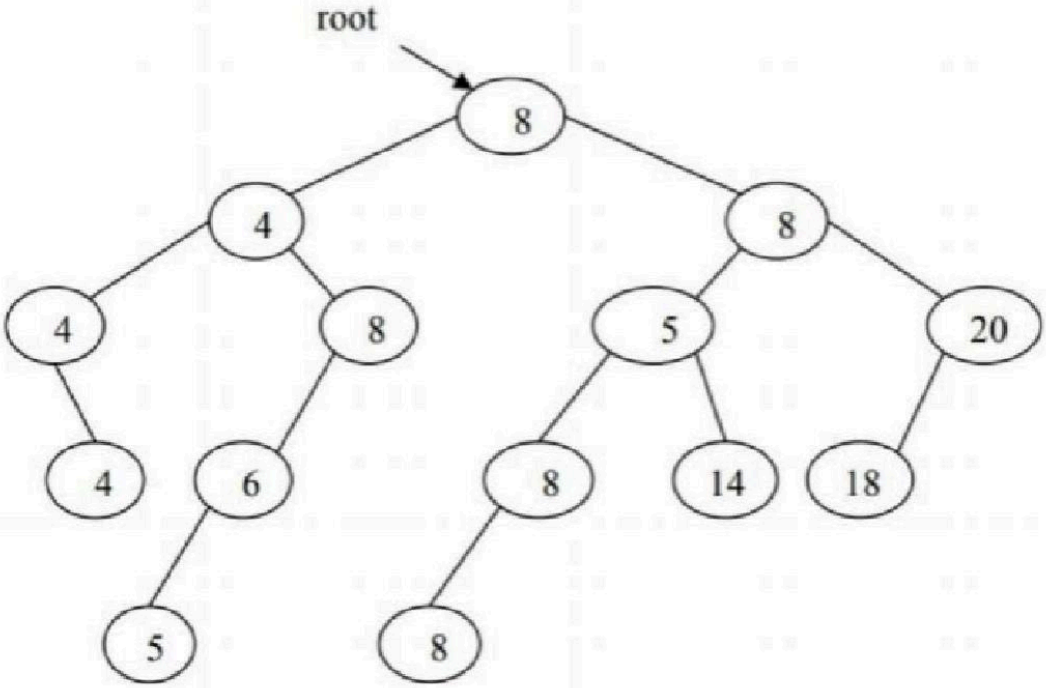
המחלקה BinaryTree מאגדת בתוכה שיטות סטטיות לטיפול בעץ בינרי. התבוננו בשיטה הבאה וענו על השאלות שאחריה:

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

שימו לב, אכן יש בשיטה הרבה סוגריים והביטויים הלוגיים מורכבים, אך אין בה טעות. בדקו היטב מה מוחזר בכל אחד מהתנאים.

שאלה 2 המשך

בהינתן **העץ הבינרי** הבא:



1) מה יודפס כתוצאה מהפקודה:

```
System.out.println (BinaryTree.f(root));
```

כתבו את תשובתכם כאן (

2) מהם השינויים **המינימליים** שעלינו לבצע **בעץ הנתון לעיל** כדי שהשיטה f תחזיר תוצאה אחרת מזו שהודפסה בסעיף 1. **(שינויים בעץ ולא בשיטה).**

כתבו את תשובתכם כאן (

3) מה מבצעת השיטה f באופן כללי, כאשר היא מקבלת שורש לעץ בינרי?

כתבו את תשובתכם כאן (