

```

class WishesTree
{
    private int n;
    private BinNode<Node<Integer>> numTree;    // עץ של רשימות
    // פעולה בונה, פעולות מאחזרות וקובעות
    public void addNumber(int num)
        // הפעולה מוסיפה את num לכל הרשימות בעצים שהוא לא נמצא בהן, בטווח 1-n
    {
        if (num >= 1 && num <= this.n && this.numTree != null)
        {
            Node<Integer> p = this.numTree.getValue();
            boolean ok = false;
            while (p != null && !ok)
            {
                if (p.getValue() == num)    // בודק האם איבר נמצא ברשימה
                    ok = true;
                p = p.getNext();
            }
            if (!ok)    // האיבר לא נמצא ברשימה
                this.numTree.setValue(new Node<Integer>(num, this.numTree.getValue()));
            WishesTree left = new WishesTree(this.n, this.numTree.getLeft()); // יצירת עצם
                // לתת העץ השמאלי
            left.addNumber(num);    // הפעלת הפעולה על תת העץ השמאלי
            WishesTree right = new WishesTree(this.n, this.numTree.getRight()); // יצירת
                // לתת העץ הימני
            right.addNumber(num);    // הפעלת הפעולה על תת העץ הימני
        }
    }
}
// סעיף ב
public static int numAppear(BinNode<Node<Integer>> t, int num) // פעולת עזר
{
    // טענת כניסה : הפעולה מקבלת עץ של רשימות של מספרים שלמים ומספר
    // טענת יציאה : הפעולה מחזירה את מספר הופעות המספר בכל הרשימות בעץ
    if (t == null)
        return 0;
    Node<Integer> p = t.getValue();
    int count = 0;
    while (p != null)
    {
        if (p.getValue() == num)
            count++;
        p = p.getNext();
    }
    return numAppear(t.getLeft(), num) + numAppear(t.getRight(), num) + count;
}
public static Node<Integer> maxAppear(WishesTree wt)
{
    // טענת כניסה : הפעולה מקבלת עצמם מטיפוס WishesTree
}

```

Java

Java

```

// טענת יציאה : הפעולה מחזירה את רשימת המספרים שמספר החזרות שלהם בעץ הוא הגדול ביותר
int max = -1;
for (int i = 1 ; i<= wt.getN() ; i++) // מציאת מספר החזרות המקסימלי של מספר
{
    int x= numAppear(wt.getNumTree(),i);
    if (x > max)
        max = x;
}
Node<Integer> ls = null;
for (int i = 1 ; i<= wt.getN(); i++) // הכנסת המספרים שמספר חזרותיהם מקסימלי לרשימה חדשה
    if (numAppear(wt.getNumTree(),i) == max)
        ls = new Node<Integer>(i,ls);
return ls;
}

```