

```

public static Node<Item> CountListTree (BinNode<int>T, Node<int>L)
{ // הפעולה מקבלת עץ ורשימה של מספרים שלמים ומחזירה רשימה מטיפוס Item
// לכל איבר בטיפוס : ערך, מספר מופעיו ברשימה, ומספר מופעיו בעץ.
    Node<Item> lst = null;
    SearchInList (T, L, lst);
    SearchInTree (T, L, lst);
    return lst;
}

public static void SearchInList (BinNode<int>T, Node<int>L, Node<int>lst)
{ // פעולת חיפוש מספרים ברשימה
    int num;
    Node<int >pos = L;
    Node<Item> q;
    while (pos != null)
    {
        num = pos.GetValue();
        if (!IsInList(lst, num)) // אם האיבר לא קיים ברשימה החדשה
        {
            q = new Node<Item>(new Item(num, CountInList(L, num), CountInTree(s, num)),lst);
            lst = q;
            pos = pos.GetNext();
        }
    }
}

public static void SearchInTree (BinNode<int>T, Node<int>L, Node<int>lst)
{ // פעולת חיפוש מספרים בעץ
    if (T != null)
    {
        if (!IsInList(lst, T.GetValue()))
            lst= new Node<Item>(new Item(s.Top(), CountInList(L, s.Top()),
            CountInTree(s,s.Top()),lst);

        SearchInTree (T.GetLeft(), L, lst);
        SearchInTree (T.GetRight(), L, lst);
    }
}

public static bool IsInList (Node<Item> lst, int num)
{ // הפעולה מקבלת רשימה ומספר ומחזירה אמת אם המספר נמצא ברשימה ושקר אחרת
    Node<int>pos = lst;
    while (pos != null)
    {
        if (pos.GetValue().GetNum() == num)
            return true;
        pos = pos.GetNext();
    }
    return false;
}

public static int CountInList (Node<int>L, int num)

```

```

    הפעולה מקבלת רשימה מקושרת ומספר ומחזירה כמה פעמים המספר מופיע ברשימה
    { //
      Node<int>pos = L;
      int count = 0;
      while (pos != null)
      {
        if (pos.GetValue() == num)
          count++;
        pos = pos.GetNext();
      }
      return count;
    }
private static int CountInTree (BinNode<int>T, int num)
{ // הפעולה מקבלת עץ בינארי ומספר ומחזירה כמה פעמים המספר מופיע בעץ
  if (T == null)
    return 0;
  if (T.GetValue() == num)
    return (1 + CountInTree(T.GetLeft(), num) + CountInTree(T.GetRight(), num));
  return (CountInTree(T.GetLeft(), num) + CountInTree(T.GetRight(), num));
}

```