המחלקה Item:

```
class Item
{
   private int num;
   private int countInList;
   private int countInStack;

   public Item(int num, int countInList,int countInStack)
   {
        this.num = num;
        this.countInList = countInList;
        this.countInStack = countInStack;
   }

   int GetNum()
   {
      return this.num;
   }
   public string ToString()
   {
      return "num="+this.num+" CountInList "+this.countInList+" CountInStack "+this.countInStack;
   }
}
```

טבלת פעולות:

שם הפעולה	תיאור הפעולה
IsInList	פעולה הבודקת אם איבר נמצא ברשימה ומחזירה true/false בהתאם.
SearchInStack	פעולה הסורקת את המחסנית, ובודקת אם האיבר הנוכחי נמצא ברשימה החדשה או לא. אם האיבר לא נמצא הפעולה מונה את מספר המופעים של המספר ברשימה ובמחסנית(בעזרת פעולות עזר), בונה עצם חדש ומכניסה אותו לרשימה.
SearchInList	פעולה הסורקת את הרשימה המקורית, ובודקת אם האיבר הנוכחי נמצא ברשימה החדשה או לא. אם האיבר לא נמצא הפעולה מונה את מספר המופעים של המספר ברשימה ובמחסנית (בעזרת פעולות עזר), בונה עצם חדש ומכניסה אותו לרשימה.
CountListStack	**הפעולה הראשית**. יוצרת רשימה חדשה, ומזמנת את הפעולות של סריקת אברי הרשימה וסריקת אברי המחסנית. הפעולה מחזירה את הרשימה המבוקשת.
CountInList	פעולה המחזירה את מספר הפעמים שאיבר מופיע ברשימה
CountInStack	פעולה המחזירה את מספר הפעמים שאיבר מופיע במחסנית

```
public static Node<Item> CountListStack(Stack<int> s,Node<int> L)
{ // Item הפעולה מקבלת מחסנית  ורשימה של מספרים שלמים ומחזירה רשימה מטיפוס.
  // לכל איבר בטיפוס: ערך, מספר מופעיו ברשימה, ומספר מופעיו במחסנית.
   Node<Item> lst = null;
   SearchInList(s, L, lst);
   SearchInStack(s, L, lst);
   return lst;
}
public static void SearchInList(Stack<int> s, Node<int> L, Node<Item>lst)
{ // פעולת חיפוש מספרים ברשימה
   int num;
```

```
    Node<int> pos = L;
    Node<Item> q;
    while  (pos != null)
    {
       num = pos.GetValue();
       if (!IsInList(lst, num)) // האם מספר לא נמצא ברשימה
       {
          q = new Node<Item>(new Item(num, CountInList(L, num), CountInStack(s, num)),lst);
          lst = q;
       }
       pos = pos.GetNext();
    }
}
public static void SearchInStack(Stack<int> s, Node<int> L, Node<Item> lst)
{ // פעולת חיפוש מספרים במחסנית
    Stack<int> temp = new Stack<int>();
    Node<Item> q;
    while (!s.IsEmpty())
    {
       if (!IsInList(lst, s.Top()))
       {
          q = new Node<Item>(new Item(s.Top(), CountInList(L, s.Top()), CountInStack(s,s.Top())),lst);
          lst = q;
       }
       temp.Push(s.Pop());
    }
    while (!temp.IsEmpty())//החזרת המחסנית למקוריותה
       s.Push(temp.Pop());
}
public static bool IsInList(Node<Item> lst, int num)
{ // הפעולה מקבלת רשימה ומספר ומחזירה אמת אם המספר נמצא ברשימה ושקר אחרת
    Node<Item> 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;
```

```
}
public static int CountInStack(Stack<int> s, int num)
{ // הפעולה מקבלת מחסנית ומספר ומחזירה כמה פעמים המספר מופיע במחסנית
   int count = 0;
   Stack<int> temp = new Stack<int>();
   while (!s.IsEmpty())
   {
      if (s.Top() == num)
         count++;
      temp.Push(s.Pop());
   }
   while (!temp.IsEmpty())
      s.Push(temp.Pop());
    return count;
}
```