

```

public static void insertToQueueStack(Stack<Queue<Integer>> SQ, int num)
{
    Stack<Queue<Integer>> temp = new Stack<Queue<Integer>>(); // מחסנית עזר
    boolean inserted = false;
    while (!s.isEmpty()) // הפוך העברת אברי המחסנית למחסנית החדשה בסד
        temp.push(SQ.pop());
    while (!temp.isEmpty())
    {
        if (!exist(temp.top(), num) && !inserted)
        { // המספר לא נמצא בתור ועדיין לא הוכנס למחסנית התורים
            temp.top().insert(num);
            inserted = true;
        }
        SQ.push(temp.pop());
    }
    if (!inserted) // אם אחרי הסריקה המספר עדיין לא הוכנס
    {
        Queue<Integer> q = new Queue<Integer>();
        q.insert(num);
        SQ.push(q); // הוספת המחסנית החדשה לראש מחסנית התורים
    }
}

public static boolean exist(Queue<Integer> q, int x)
{ // טענת כניסה: הפעולה מקבלת תור של מספרים שלמים ומספר שלם
    // טענת יציאה: הפעולה מחזירה true אם המספר נמצא בתור, או false אחרת
    Queue<Integer> temp = new Queue<Integer>(); // תור עזר
    boolean exist = false;
    while (!q.isEmpty())
    {
        if (q.head() == x)
            exist = true;
        temp.insert(q.remove());
    }
    while (!temp.isEmpty())
        q.insert(temp.remove()); // החזרת איברים לתור בפרמטר
    return exist;
}

```