Disk Tower Construction using Recursion in Java

This Java program simulates the construction of a disk tower using recursion. Disks are received one per day, and the largest unplaced disk must be placed before any smaller disks. The program processes disks recursively and outputs the placement status each day.

```
import java.util.*;
public class DiskTowerRecursion {
static int expectedDisk;
static Stack stack = new Stack<>();
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter total number of disks: ");
int n = sc.nextInt();
int[] disks = new int[n];
System.out.println("Enter disk sizes received each day:");
for (int i = 0; i < n; i++) {
disks[i] = sc.nextInt();
}
expectedDisk = n;
System.out.println("\nDisk placement order each day:");
processDisks(disks, 0, n);
}
static void processDisks(int[] disks, int day, int totalDays) {
if (day == totalDays) return;
System.out.print("Day " + (day + 1) + ": ");
int currentDisk = disks[day];
stack.push(currentDisk);
boolean placed = false;
while (!stack.isEmpty() && stack.contains(expectedDisk)) {
stack.removeElement(expectedDisk);
System.out.print(expectedDisk + " ");
expectedDisk--;
placed = true;
}
if (!placed) {
System.out.print("No disk placed today");
}
System.out.println();
processDisks(disks, day + 1, totalDays);
}
}
```