🏆 **LeaderBoard & Yesterday's Solution(/faces/candidate/leaderboarddailychallenge.xhtml?RT=DAILYCHALLENGE)**

**Daily Challenge** **Happy Coding from necse** **SkillRack**

## Electronic Display - Maximum Value

In a four-digit electronic display, each digit is displayed using **seven** segments. The program must accept an integer **N** representing the number of segments to display an integer **X**. The program must print the maximum possible four-digit integer X that can be displayed using N segments as the output.
The number of segments required to display each digit is given below.
0 - 6
1 - 2
2 - 5
3 - 5
4 - 4
5 - 5
6 - 6
7 - 3
8 - 7
9 - 6

**Boundary Condition(s):**
8 <= N <= 28

**Input Format:**
The first line contains N.

**Output Format:**
The first line contains X.

**Example Input/Output 1:**
Input:
13

Output:
9711

Explanation:
Here N = 13.
The maximum possible four-digit integer that can be displayed using **13** segments is **9711** (9+7+1+1 = 13).

**Example Input/Output 2:**
Input:
10

Output:
7711

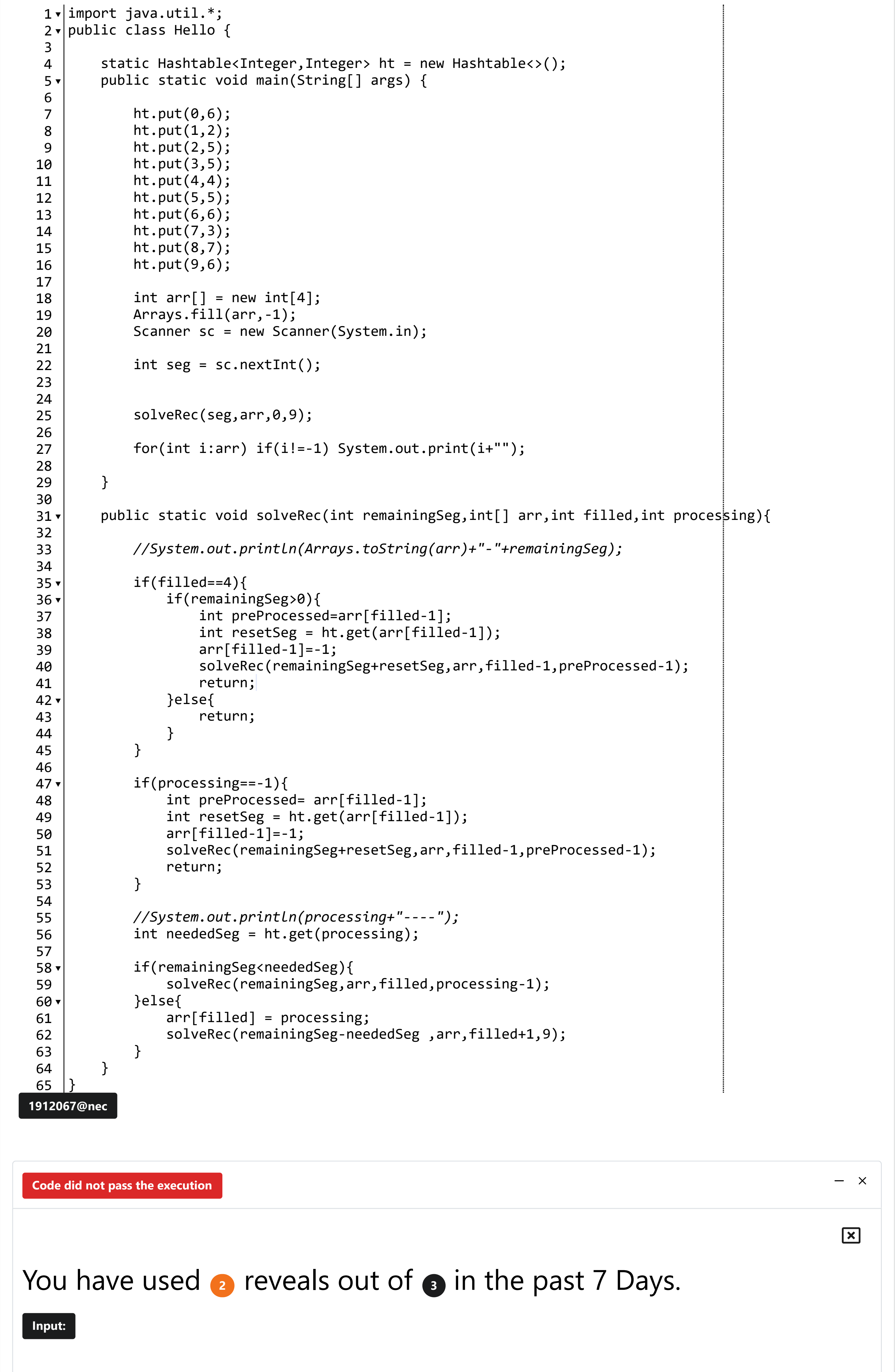
**Example Input/Output 3:**
Input:
18

Output:
9977

**Max Execution Time Limit: 50 millisecs**

Ambiance

Java ( 12.0)



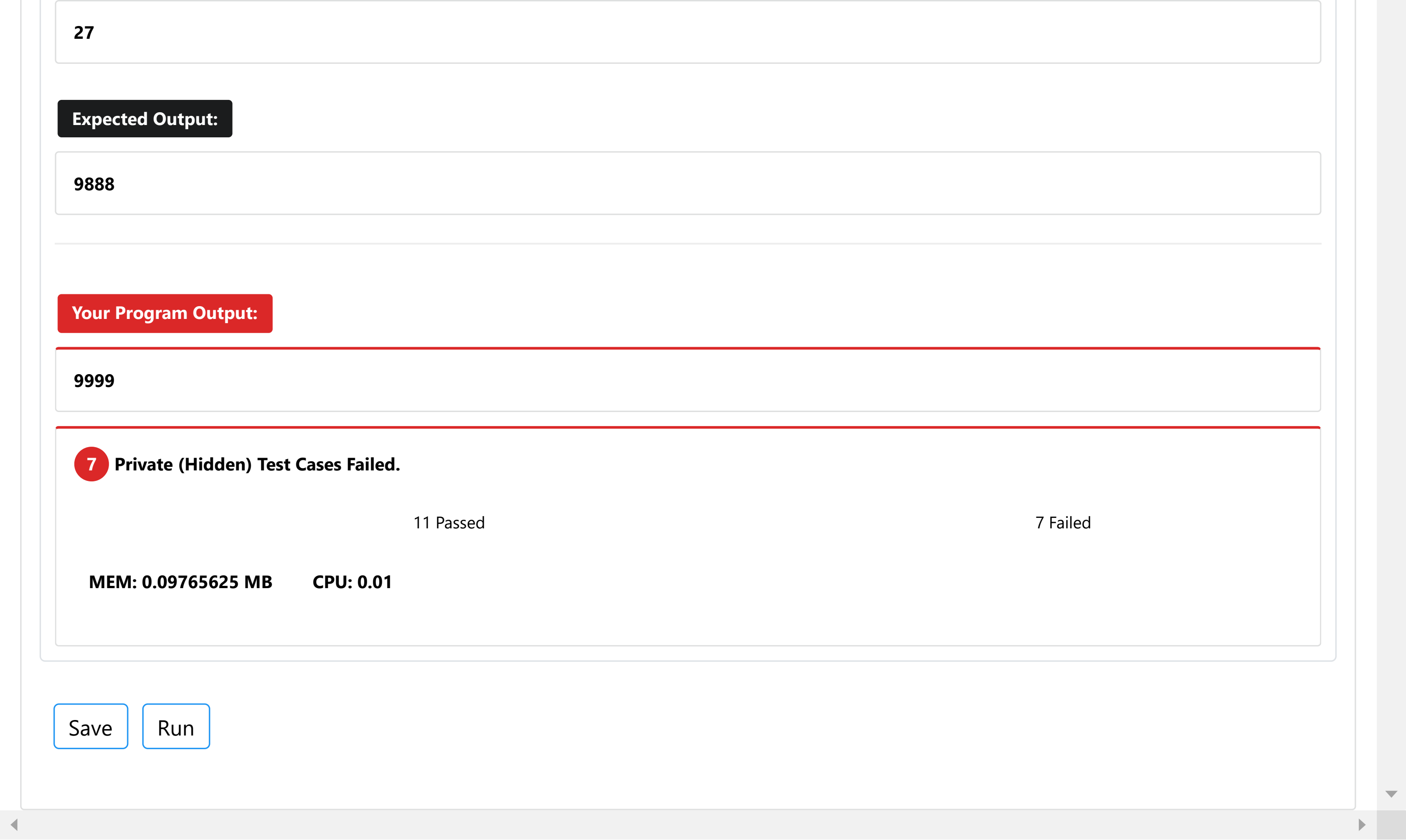
```java
import java.util.*;
public class Hello {

    static Hashtable<Integer,Integer> ht = new Hashtable<>();
    public static void main(String[] args) {

        ht.put(0,6);
        ht.put(1,2);
        ht.put(2,5);
        ht.put(3,5);
        ht.put(4,4);
        ht.put(5,5);
        ht.put(6,6);
        ht.put(7,3);
        ht.put(8,7);
        ht.put(9,6);

        int arr[] = new int[4];
        Arrays.fill(arr,-1);
        Scanner sc = new Scanner(System.in);

        int seg = sc.nextInt();


        solveRec(seg,arr,0,9);

        for(int i:arr) if(i!=-1) System.out.print(i+"");

    }

    public static void solveRec(int remainingSeg,int[] arr,int filled,int processing){

        //System.out.println(Arrays.toString(arr)+"-"+remainingSeg);

        if(filled==4){
            if(remainingSeg>0){
                int preProcessed=arr[filled-1];
                int resetSeg = ht.get(arr[filled-1]);
                arr[filled-1]=-1;
                solveRec(remainingSeg+resetSeg,arr,filled-1,preProcessed-1);
                return;
            }else{
                return;
            }
        }

        if(processing==-1){
            int preProcessed= arr[filled-1];
            int resetSeg = ht.get(arr[filled-1]);
            arr[filled-1]=-1;
            solveRec(remainingSeg+resetSeg,arr,filled-1,preProcessed-1);
            return;
        }

        //System.out.println(processing+"----");
        int neededSeg = ht.get(processing);

        if(remainingSeg<neededSeg){
            solveRec(remainingSeg,arr,filled,processing-1);
        }else{
            arr[filled] = processing;
            solveRec(remainingSeg-neededSeg ,arr,filled+1,9);
        }
    }
}
```

1912067@nec

Code did not pass the execution

You have used 2 reveals out of 3 in the past 7 Days.

Input:



```
27
```

**Expected Output:**

```
9888
```

**Your Program Output:**

```
9999
```

7 **Private (Hidden) Test Cases Failed.**

11 Passed 7 Failed

**MEM: 0.09765625 MB** **CPU: 0.01**