**Jarvis and Seven Segments**

Attempted by: **742**

/

Accuracy: **47%**

/

Maximum Score: **10**

/

9 Votes

Tag(s):

Ad-Hoc, Very-Easy, Very-Easy, ad-hoc

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

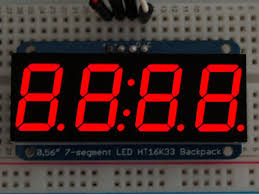
**ANALYTICS**

All over the world, peoples are working on energy solution. It would be a tough time for our next generation to survive if we don’t think about solution. Tony stark is working on a new project and wants to display his project using “***seven segment display - concept***”. Tony Stark gave Jarvis a task to find a number from his Favorite list of number for which the energy consumption is lowest.

(Assuming that for a digit to represent Tony stark is using 7 bulbs and ***only those bulbs light up which are required to represent a number and rest other would be completely off.***)

Help Jarvis and conserve energy.

Seven segment display - <https://en.wikipedia.org/wiki/Seven-segment_display>



**Input:**  
First line will contain the number of test cases and for every test case first line will contain length of favorite list and the second line for a test case will contain n numbers

**Output**:  
For every test case print the answer. If there exist more than 1 numbers for which same number of bulbs are required than output the number which occurs first in the Favorite list.

***Constraints:***  
**Test cases< 10  
A[i] < 10^6  
Size of list < 10^5**

**SAMPLE INPUT**

1

5

1 2 3 4 5

**SAMPLE OUTPUT**

1

**Explanation**

Number 1 needs only two bulbs to represent.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/jarvis-and-seven-segments-1/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int obtenerNumBombillas(string n)

{

Dictionary<char, int> diccio = new Dictionary<char, int>();

diccio['0'] = 6;

diccio['1'] = 2;

diccio['2'] = 5;

diccio['3'] = 5;

diccio['4'] = 4;

diccio['5'] = 5;

diccio['6'] = 6;

diccio['7'] = 3;

diccio['8'] = 7;

diccio['9'] = 6;

int total\_bombillas = 0;

for (int i = 0; i < n.Length; i++)

{

total\_bombillas += diccio[n[i]];

}

return total\_bombillas;

}

static void Main(string[] args)

{

// int[] numbers = Array.ConvertAll("1 2 3 4 5".Split(' '), e => int.Parse(e));

// string[] numbers = "1 2 3 4 5".Split(' ');

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine());

string[] numbers = Console.ReadLine().Split(' ');

string min\_num = "";

int min\_bombillas = int.MaxValue;

for (int i = 0; i < numbers.Length; i++)

{

int bombillas = obtenerNumBombillas(numbers[i]);

if (bombillas < min\_bombillas)

{

min\_bombillas = bombillas;

min\_num = numbers[i];

}

}

Console.WriteLine(min\_num);

}

Console.ReadLine();

}

}

}