

A. Costly Binary Search

B. Campinatorics

C. Pretty Good Proportion

D. Taking Over The World

E. Merlin QA

F. Crane Truck

Contest Analysis

Questions asked

Submissions

Costly Binary Search

8pt	Not attempted 20/25 users correct (80%)
19pt	Not attempted 16/17 users correct (94%)

Campinatorics

6pt	Not attempted 25/25 users correct (100%)
21pt	Not attempted 23/25 users correct (92%)

Pretty Good Proportion

5pt	Not attempted 26/26 users correct (100%)
22pt	Not attempted 10/18 users correct (56%)

Taking Over The World

7pt	Not attempted 20/21 users correct (95%)
29pt	Not attempted 3/4 users correct (75%)

Merlin QA

8pt	Not attempted 14/19 users correct (74%)
30pt	Not attempted 4/8 users correct (50%)

Crane Truck

8pt	Not attempted 2/3 users correct (67%)
37pt	Not attempted 0/1 users correct (0%)

Top Scores

Gennady.Korotkevich	155
rng..58	134
bmerry	104
tczajka	96
vepifanov	96
peter50216	96
tkociumaka	96
linguo	92
simonlindholm	77
pashka	76

Problem A. Costly Binary Search

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
8 points

Solve A-small

Large input
19 points

Solve A-large

Problem

You were asked to implement arguably the most important algorithm of all: binary search. More precisely, you have a sorted array of objects, and a new object that you want to insert into the array. In order to find the insertion position, you can compare your object with the objects in the array. Each comparison can return either "greater", meaning that your object should be inserted to the right of the compared object, or "less", meaning that your object should be inserted to the left of the compared object. For simplicity, comparisons never return "equal" in this problem. It is guaranteed that when your object is greater than some object in the array, it is also greater than all objects to the left of that object; similarly, when your object is less than some object of the array, it is also less than all objects to the right of that object. If the array has n elements, there are $n+1$ possible outcomes for your algorithm.

In this problem, not all comparisons have the same cost. More precisely, comparing your object with i -th object in the array costs a_i , an integer between 1 and 9, inclusive.

What will be the total cost, in the worst case, of your binary search? Assume you follow an optimal strategy and try to minimize the total cost in the worst case.

Input

The first line of the input gives the number of test cases, T . T lines follow. Each of those lines contains one sequence of digits describing the comparison costs a_i for one testcase. The size of the array n is given by the length of this sequence.

Output

For each test case, output one line containing "Case $\#x$: y ", where x is the test case number (starting from 1) and y is the total binary search cost in the worst case.

Limits

$1 \leq T \leq 50$.
All digits are between 1 and 9, inclusive.
There are no spaces between digits on one line.

Small dataset

$1 \leq n \leq 10^4$.

Large dataset

$1 \leq n \leq 10^6$.

Sample

Input	Output
4	Case #1: 2
111	Case #2: 3
1111	Case #3: 3
1111111	Case #4: 10
1111119	

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform