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 $\mathbf{a_i}$, 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 $\mathbf{a_i}$ 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

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

Small dataset

$1 \le \mathbf{n} \le 10^4$.

Large dataset

$1 \le \mathbf{n} \le 10^6$.

Sample

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