Permutation Signature

Problem Statement

The *signature* of a permutation is a string that is computed as follows:

for each pair of consecutive elements of the permutation, write down the letter 'I' (increasing) if the second element is greater than the first one, otherwise write down the letter 'D' (decreasing).

For example, the signature of the permutation {3,1,2,7,4,6,5} is "DIIDID".

Your task is to reverse this computation: You are given a String **signature** containing the signature of a permutation. Find and print the lexicographically smallest permutation with the given signature. It is guaranteed that there must be a valid permutation exists.

Notes

-For any positive integer N, a permutation of N elements is a sequence of length N that contains each of the integers 1 through N exactly once.

-To compare two permutations A and B, find the smallest index i such that A[i] and B[i] differ. If A[i] < B[i], we say that A is lexicographically smaller than B, and vice versa.

Input

There are multiple test cases for this problem. For each test case, a string **signature** will be given. **signature** will contain between 1 and 50 characters, inclusive. Each character in **signature** will be either 'I' or 'D'.

The input ends with EOF

Ouput

For each test case, output the number of test case first (Case #X:), then followed by the sequence of lexicographically smallest permutation**,** output the sequence with **exactly a space separating each element, no space after the sequence**.

Sample Input and Output

Input	Ouput
IIIII	**Case #1:** 1 2 3 4 5 6
DI	**Case #2: 2 1 3**
IIIID	**Case #3:** 1 2 3 4 6 5
DIIDID	**Case #4:** 2 1 3 5 4 7 6