Problem Statement (PermutationSignature)

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 return the lexicographically smallest permutation with the given signature. If no such permutation exists, return an empty int[] instead.

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.

Constraints

-**signature** will contain between 1 and 50 characters, inclusive.-Each character in **signature** will be either 'I' or 'D'.

Examples

0)

```
"IIIII"
Returns: {1, 2, 3, 4, 5, 6 }
```

1)

```
"DI"
Returns: {2, 1, 3 }
```

There are two permutations with this signature: {3,1,2} and {2,1,3}. You must return the lexicographically smaller one.

2)

```
"IIIID"
Returns: {1, 2, 3, 4, 6, 5 }
```

3)

```
"DIIDID"
Returns: {2, 1, 3, 5, 4, 7, 6 }
```

This is the signature from the problem statement. Note that the correct answer is not the permutation from the problem statement.