

H: String Reversals

Consider a string of binary digits. It consists of a sequence of 0's and 1's in some arbitrary order. Now let's sort the digits so that all the 0's are on the left and the 1's are on the right. We will do this by reversing segments of the string and reattaching them back in the same place. A reversal in the segment may mean that one end of the strand becomes separated (a single break), becomes turned around, and reconnects by its other end. It may also mean that a segment inside the original string becomes separated (two breaks), is turned around, and rejoins the other two pieces exactly where it had broken.

You will be presented with the initial string. Your task is to determine the minimum number of reversals that would be needed to sort the digits.

Input

Input may consist of multiple cases. Each case consists of one line containing the string of binary digits to sort. The string will consist of up to 1000 digits, and there may be arbitrary white space before and/or after the string. The final case will consist of a string of all 0's (which you should solve!) and there will be nothing more to process after that final case.

Output

For each case, display the case number followed by the answer, formatted as in the sample. Delimiters should be single spaces.

Sample Input

```
1100
  1010
 11001100
00000
```

Sample Output

```
Case 1: 1
Case 2: 2
Case 3: 2
Case 4: 0
```
