Can You Make It Empty 3

Let us introduce an algorithm with a function **CanEmpty()** which takes a string P as a parameter and return **TRUE** if it is possible to make P empty, otherwise return **FALSE**.

String P consists of only 0 and 1. The pseudo-code implementation of **CanEmpty()** function is as follows.

```
bool CanEmpty(String P)
{
    while (P has at least one substring 100)
    {
       Chose any one substring 100 in P and delete it.
    }
    if (P is empty) return TRUE;
    else return FALSE;
}
```

Now you are given a string S consisting of 0 and 1, you have to find the length of longest substring of **S** that can be made empty applying **CanEmpty()** algorithm.

As for example, let S=101**110000**0**100**

S has only two sub-strings (bold) which can be made empty applying **CanEmpty()** algorithm.

The first substring will have the delete- sequence in **CanEmpty()** function :

1<u>100</u>00-><u>100</u>->empty

The second substring will have the delete-sequence in **CanEmpty()** function:

<u>100</u>->empty

The length of first substring is 6 and second is 3. So, the required answer is 6.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains a string S. The size of string is at most **200000**.

Output

For each test case, print the case number and required answer.

Sample Input	Output for Sample Input
2 1011100000100 111011	Case 1: 6 Case 2: 0

Problem Setter: Md Abdul Alim, Dept. of Computer Science, Bangladesh University of Business & Technology