Get Eliminated

Problem Statement

You are given a binary string S of size N. In one operation, you can remove S_i and S_{i+1} if S_i is equal to S_{i+1} and S_i is equal to 1 for each i and $0 \leq i < |S| - 1$. You need to perform this operation until no further operations can be done. Each operation must be performed on the new string. You need to find the maximum number of continuous $0's$ in the final string.

Input Format

- First line will contain T, the number of test cases.
- Each line of the test case will contain the string S.

Constraints

1. $1 \leq T \leq 10^3$
2. $1 \leq |S| \leq 10^5$. Here $||$ means the length of the string.

Output Format

- Output the maximum number of continuous 0's.

Sample Input 0

```
5
01111001010111100110
01101110
1110000111
11
000
```

Sample Output 0

```
4
2
4
0
3
```

Explanation 0

```
In the first test case, the final string will look like 0001010000 where the number of continuous 0's are 3,1
and 4. So, the answer is 4 which is maximum.
```

More

C++20

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code