- **EASY**

- **Max Score: 30 Points**

-

Minimum Number of Flips to Make the Binary String Alternating

Ankit has been given a binary string `str` containing either `0` or `1`. A binary string is called beautiful if it contains alternating `0s` and `1s`.

For Example: '0101', '1010', '101', '010' are beautiful strings.

He wants to make `str` beautiful by performing some operations on it. In one operation, Ankit can convert `0` into `1` or vice versa.

Your task is to determine the minimum number of operations Ankit should perform to make `str` beautiful.

Input Format

The only line of input contains a binary string `str`.

You need to complete `makeBeautiful` function which contains string `str` and return minimum number of operation required

Output Format

Print the minimum operations needed to make `str` beautiful.

Example 1

Input

0000

Output

2

Explanation

The two beautiful strings that can be formed from the given 'str' are "1010" and "0101".

Ankit can transform 'str' to "1010" by performing the following operations:

Replace '0' at index 0 by '1' and replace '0' at index 2 by '1'.

Ankit can transform 'str' to "0101" by performing the following operations:

Replace '0' at index 1 by '1' and replace '0' at index 3 by '1'.

The minimum number of operations in transforming 'str' to either of the two beautiful strings is 2.

Example 2

Input

1010

Output

0

Explanation

Given 'str' is already beautiful so the minimum number of operations required is 0.

Constraints

2 <= |str| <= 10^5

str[i] = '1' or '0', where '|str|' denotes the length of 'str'.

- **Strings**

- **Sliding Window**

- **Greedy**

- **DP**

My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
    public static void main (String[] args) throws java.lang.Exception
    {
        //your code here
    //here ya to 1 se start ya fir 0 se
    Scanner s=new Scanner(System.in);
    String st=s.next();
    //if start with 0
    int sum1=0;
     //if start with 1
    int sum2=0;
    for(int i=0;i<st.length();i++)
      {
      char ch =st.charAt(i);
      int r=i%2;
```

```
      if(r==0 && ch=='1') sum1+=1;
      if(r==0 && ch=='0') sum2+=1;
       if(r==1 && ch=='0') sum1+=1;
       if(r==1 && ch=='1') sum2+=1;
      }
    int max=sum1<sum2?sum1:sum2;
    System.out.print(max);




      }
  }
```