

Question 1: Backspace String Compare

Given two strings `S` and `T`, return if they are equal when both are typed into empty text editors. `#` means a backspace character.

Sample Input:

```
S = "ab#c", T = "ad#c"
```

Sample Output:

```
True
```

Sample Input:

```
S = "ab##", T = "c#d#"
```

Sample Output:

```
True
```

Sample Input:

```
S = "a#c", T = "b"
```

Sample Output:

```
False
```

Question 2: First Unique Character in a String

Given a string, find the first non-repeating character in it and return it's index. If it doesn't exist, return -1.

Sample Input:

```
s = "liveshow"
```

Sample Output:

0

Sample Input:

```
s = "loveliveshow"
```

Sample Output:

5

Question 3: Longest substring without repeating characters

Given a string, find the length of the **longest substring** without repeating characters.

Sample Input:

abcabcbb

Sample Output:

3

Sample Input:

bbbbbb

Sample Output:

1

Sample Input:

pwwkew

Sample Output:

3

Question 4: Even or Odd String?

A string is a palindrome if and only if it reads the same forwards and backwards.

A substring of a string **s** is a nonempty sequence of consecutive characters from **s**

Define a string to be *odd* if and only if all substrings of the string that are palindromes have odd length.

Given a string **s**, determine if it is *odd*.

Print on a single line, if the string is odd, `Odd`; if not, `Even`.

Sample Input:

Amanaplanacanalpanama

Sample Output:

Odd

Sample Input:

Madamimadam

Sample Output:

Odd

Sample Input:

Annamymfriend

Sample Output:

Even

Question 5: Permutation in String

Given two strings **s1** and **s2**, write a function to return true if **s2** contains the permutation of **s1**. In other words, one of the first string's permutations is the **substring** of the second string.

Sample Input:

```
s1 = "ab" s2 = "eidbaooo"
```

Sample Output:

True

Sample Input:

```
s1= "ab" s2 = "eidboaoo"
```

Sample Output:

False