Sample Input 0

```
00110011
```

Sample Output 0

```
6
```

0 0 1 1 1 →

→

Odd length → substring

i = 2

0 0 1 1 0 0 1 1 → ✗

i = 1 ; counter1 = ✗ 2

Ch1 ⇒ '0'

0 0 1 1 0 0 1 1

while (i < substring.length) && ch1 == substring.ch(i))

counter1 = 2
counter2 = 2

{
  i++ ;
  countch1++ ;
}

1 1 0 6

0 → 1, 2

(i == substring.length)
return false ;

0 0 1

0 0 1 1 0

ch2 = '1' , countch2 = ✗ 2 ; i += 1 ;

0 0 1 1 0 0

0 0 1 1 0 0

if, (i == substring.length) && (countch1 == countch2)
return true

while (i < substring.le && ch2 == substring.char(i))

{
  i++ ;
  charbecomzero ;
}

return false ;

Reverse only letters

Given a string s, reverse the string according to the following rules:

- All the characters that are not English letters remain in the same position.
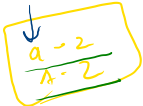- All the English letters (lowercase or uppercase) should be reversed.
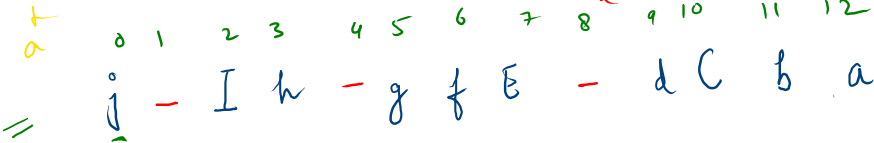
Print s after reversing it.

Sample Input 0

```
a-bC-dEf-ghIj
```

Sample Output 0

```
j-Ih-gfE-dCba
```


a - 2
A - 2

a - b C - d E f - g h I j

while l left by w

j - I h - gf E dC b a

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

j - I h - g f E - d C b a

to ChonKong

other ch = our (G,T)

or our) =

tn → left position = 0 1 2 3 4 5 6

fn → right position = 12 11 10 9 7

```java
public static int positionFromLeftSide(String str,int left,int right){

    while(left<right){
        char ch=str.charAt(left);
        if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z')){
            return left;
        }
        left++;

    }

    return -1;
}
public static int positionFromRightSide(String str,int right,int left){

    while(left<right){
        char ch=str.charAt(right);
        if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z')){
            return right;
        }
        right--;

    }

    return -1;
}
```

```java
public static void reverseLetters(String str){
    char arr[]=str.toCharArray();

    int left=0;
    int right=str.length()-1;

    while(left<right){

        left=positionFromLeftSide(str,left,right);
        right=positionFromRightSide(str,right,left);

        if(left==-1 || right==-1){
            break;
        }

        char ch=arr[left];
        arr[left]=arr[right];
        arr[right]=ch;

        left++;
        right--;


    }

    for(int i=0;i<arr.length;i++){
        System.out.print(arr[i]);
    }
}
```
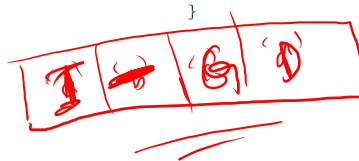
name → alex ✗
type → a

Long Pressed Name

Problem Submissions Leaderboard Discussions

Your friend is typing his name into a keyboard. Sometimes, when typing a character c, the key might get long pressed, and the character will be typed 1 or more times.

You examine the typed characters of the keyboard. Return True if it is possible that it was your friends name, with some characters (possibly none) being long pressed.

Sample Input 0

```
alex
aaleex
```

name = "alex"
typed = "aaleex"

Sample Output 0

```
true
```

--- handwritten annotations ---

name: sumeet
typed: ssuumeet

name = abcc
typed = aj

name = aab
typed = bb

name = aabb c
typed = aabb ccd

class
return true

if(i==j)
if(i≥0) && (i-1) == j

(i, i-1)

while (i-1) : j

alex
aaleex

false