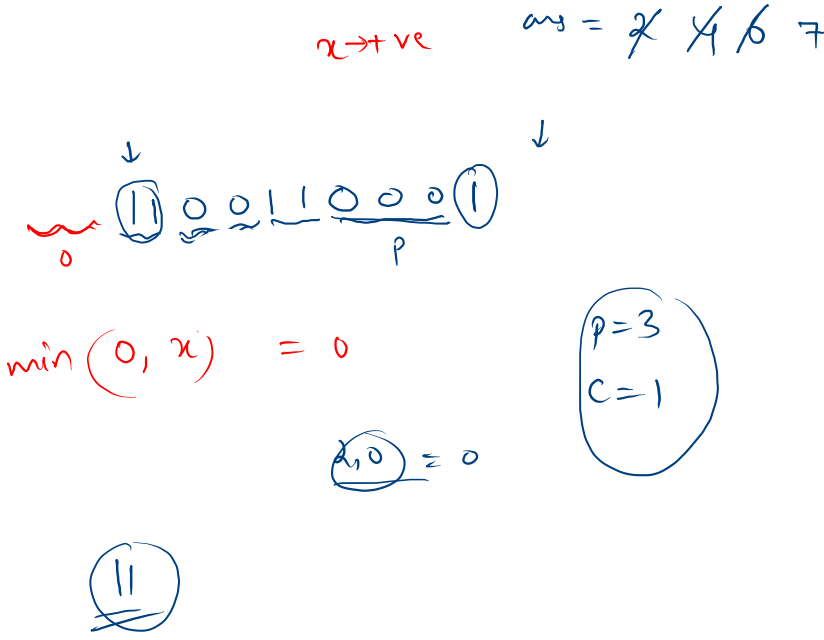
```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        int prev = 0;
        int curr = 1;
        int ans = 0;
        for(int i = 1; i < s.length(); i++){
            if(s.charAt(i) == s.charAt(i-1)){
                curr++;
            }
            else{
                ans += Math.min(curr, prev);
                prev = curr;
                curr = 1;
            }
        }
        ans += Math.min(curr, prev);
        System.out.println(ans);
    }
}
```



Merge Strings Alternatively

Problem | Submissions | Leaderboard

Take two strings as input.

Merge both the strings **alternatively**.

Note: Length of strings will be same.

s → "GEEK"

t → "AMAN"

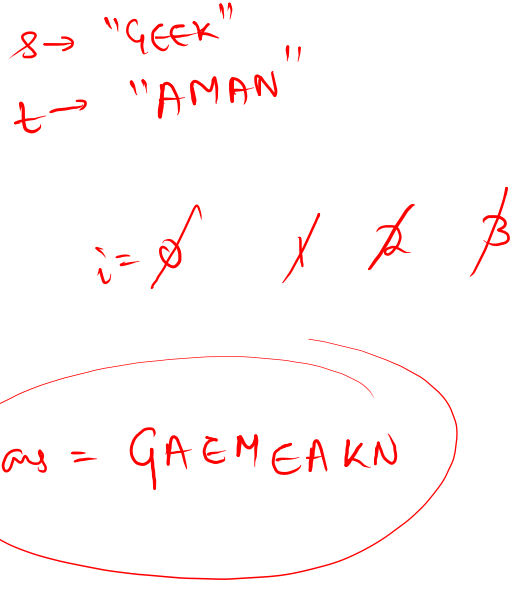
```
ans = ""
for( i=0 ⟶ s.length() )
{   ans += s.charAt(i);
    ans += t.charAt(i);
}
```

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        String t = scn.next();

        String ans = "";
        for(int i = 0; i < s.length(); i++){
            ans += s.charAt(i);
            ans += t.charAt(i);
        }
        System.out.println(ans);
    }
}
```

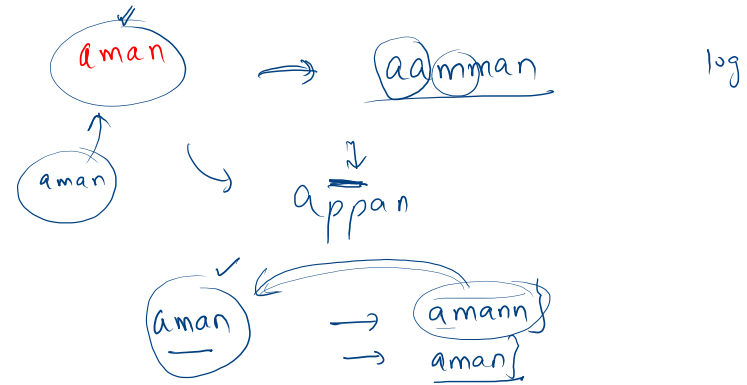


Long Pressed Name

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.



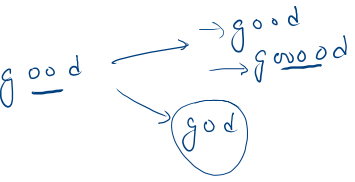

```
Input: name = "saeed", typed = "ssaaedd"
Output: false
```

Sample Input 0

```
alex
aaleex
```

Sample Output 0

```
true
```



1. $s[i] == t[j]$
   ↳ i++
   j++

2. $s[i-1] == t[j]$

s → alex

t → allexxxj

allex
allex

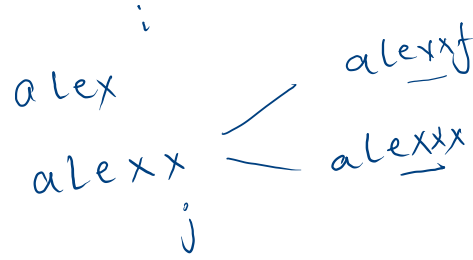
alex
aUex

n→ allex ✓

t→ alex

alex
alep
}.

```java
public static boolean longPressed(String s, String t){
    //s is actual name , t is typed name
    if(s.length() > t.length()){
        return false;
    }
    int i = 0;
    int j = 0;
    while(i < s.length() && j < t.length()){
        if(s.charAt(i) == t.charAt(j)){
            i++;
            j++;
        }
        else if(i>0 && s.charAt(i-1) == t.charAt(j)){
            j++;
        }
        else {
            return false;
        }
    }
}
```



```java
import java.io.*;
import java.util.*;

public class Solution {
    public static boolean longPressed(String s, String t){
        //s is actual name , t is typed name
        if(s.length() > t.length()){
            return false;
        }
        int i = 0;
        int j = 0;
        while(i < s.length() && j < t.length()){
            if(s.charAt(i) == t.charAt(j)){
                i++;
                j++;
            }
            else if(i>0 && s.charAt(i-1) == t.charAt(j)){
                j++;
            }
            else {
                return false;
            }
        }
        while(j < t.length()){
            if(i>0 && s.charAt(i-1) != t.charAt(j)){
                return false;
            }
            j++;
        }
        if(i <s.length()){
            return false;
        }
        else{
            return true;
        }
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        String t = scn.next();
        System.out.println(longPressed(s, t));

    }
}
```

aman

amannf

a man

a mannn

Linear Search

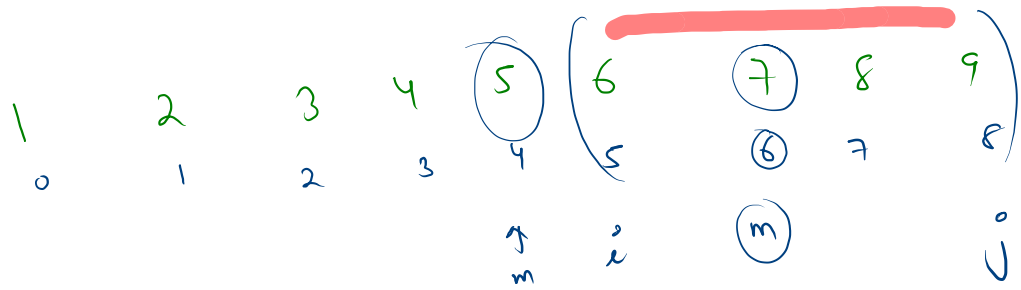
2	1	4	5	7	6	3
0	1	2	3	4	5	6

key = 6

Binary Search. → O(log n)

1. given arr should be sorted.

key = 7

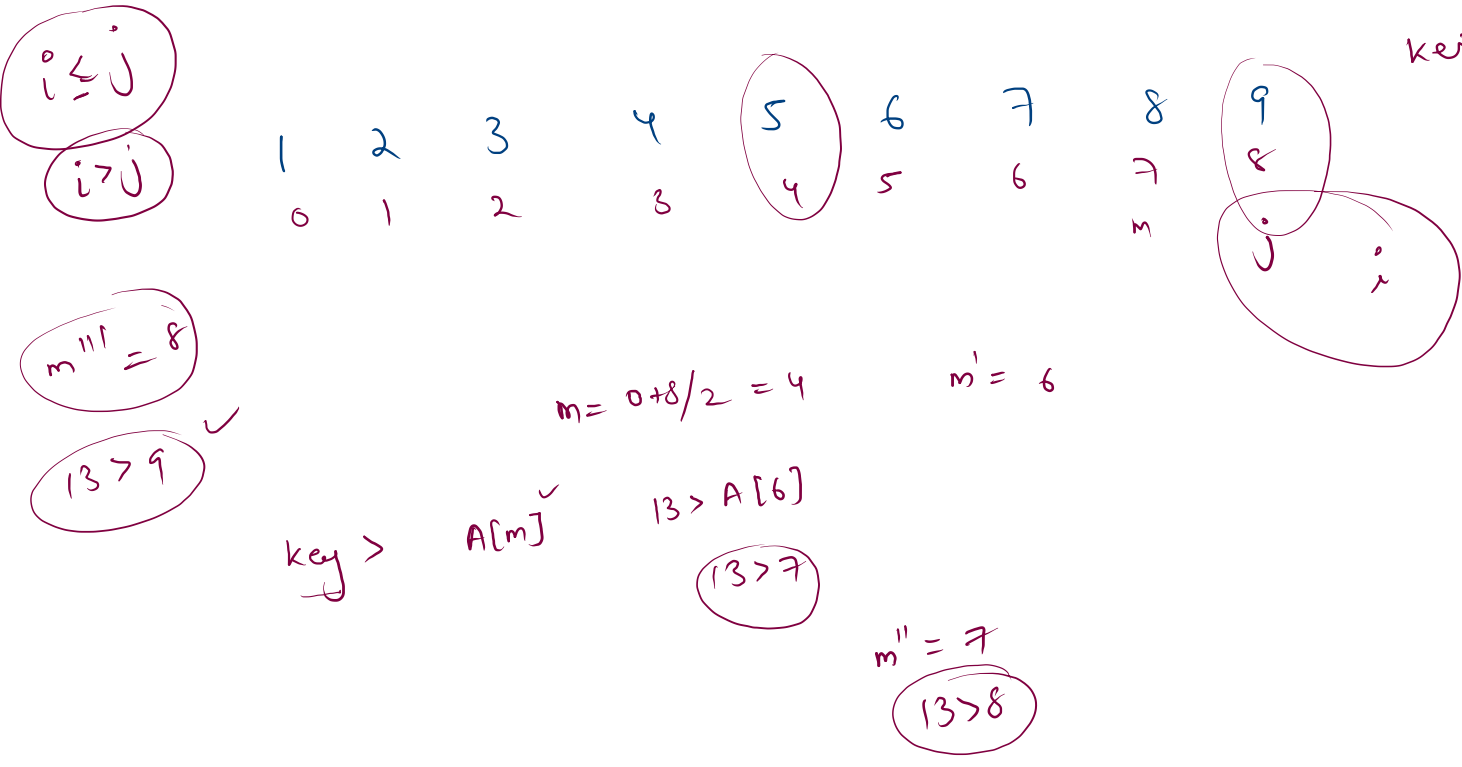


key == A[m]

key :> A[m]

m = i+j/2 = 0+8/2 = 4

m' = 5+8/2 = 6


key = 13
i ≤ j
i > j
1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8
m
j
i
m''' = 8
13 > 9
m = 0+8/2 = 4
m' = 6
key > A[m]
13 > A[6]
13 > 7
m'' = 7
13 > 8