

## 长按键入

你的朋友正在使用键盘输入他的名字 `name`。偶尔，在键入字符 `c` 时，按键可能会被 *长按*，而字符可能被输入 1 次或多次。

你将会检查键盘输入的字符 `typed`。如果它对应的可能是你的朋友的名字（其中一些字符可能被长按），那么就返回 `True`。

### 示例 1:

输入: `name = "alex", typed = "aaleex"`

输出: `true`

解释: 'alex' 中的 'a' 和 'e' 被长按。

### 示例 2:

输入: `name = "saeed", typed = "ssaaeddd"`

输出: `false`

解释: 'e' 一定需要被键入两次，但在 `typed` 的输出中不是这样。

### 示例 3:

输入: `name = "leelee", typed = "lleeelee"`

输出: `true`

### 示例 4:

输入: `name = "laiden", typed = "laiden"`

输出: `true`

解释: 长按名字中的字符并不是必要的。

### 解题思路:

根据题意能够分析得到: 字符串 `typed` 的每个字符, 有且只有两种「用途」:

- 作为 `name` 的一部分。此时会「匹配」`name` 中的一个字符
- 作为长按键入的一部分。此时它应当与前一个字符相同。

如果 `typed` 中存在一个字符, 它两个条件均不满足, 则应当直接返回 `false`; 否则, 当 `typed` 扫描完毕后, 我们再检查 `name` 的每个字符是否都被「匹配」了。

实现上, 我们使用两个下标  $i, j$  追踪 `name` 和 `typed` 的位置。

- 当 `name[i] = typed[j]` 时, 说明两个字符串存在一对匹配的字符, 此时将  $i, j$  都加 1。
- 否则, 如果 `typed[j] = typed[j - 1]`, 说明存在一次长按键入, 此时只将  $j$  加 1。

最后, 如果  $i = \text{name.length}$ , 说明 `name` 的每个字符都被「匹配」了。

```

/*bool isLongPressedName(char * name, char * typed){
    int len1=strlen(name);
    int len2=strlen(typed);

    int slow=0;
    for(int fast=0;fast<len2-1;fast++)
    {
        if(typed[fast]!=typed[fast+1])
        {
            slow++;
            typed[slow]=typed[fast+1];
        }
    }
    int len=slow+1;//更新长度
    if(len1!=len) return false;
    for(int i=0;i<len;i++)
    {
        if(typed[i]!=name[i])
        {
            return false;
        }
    }
    return true;
}
*/
bool isLongPressedName(char* name, char* typed) {
    int n = strlen(name), m = strlen(typed);
    int i = 0, j = 0;
    while (j < m) {
        if (i < n && name[i] == typed[j]) {
            i++;
            j++;
        } else if (j > 0 && typed[j] == typed[j - 1]) {
            j++;
        } else {
            return false;
        }
    }
    return i == n;
}

```