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 = "ssaaedd"
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。
```

```c
bool isLongPressedName(char * name, char * typed){

    int len1 = strlen(name);
    int len2 = strlen(typed);

    char word1[1000] = { '\0' };
    int  nums1[1000] = { 0 };
    char word2[1000] = { '\0' };
    int  nums2[1000] = { 0 };

     if (len1>len2)
     {
         return false;
     }

     char p = name[0];
     int j = 0;
     word1[0] = p;
     int i = 0;
     for (; i<len1; i++)
     {
         if (p!=name[i])  //如果不相等
         {
             p = name[i];
             j++;
             nums1[j]++;
             word1[j] = p;

         }
         else
         {
             nums1[j]++;
         }
     }

     p = typed[0];
     j = 0;
```

```
    word2[0] = p;
    i = 0;
    for (; i<len2; i++)
    {
        if (p!=typed[i])  //如果不相等
        {
            p = typed[i];
            j++;
            nums2[j]++;
            word2[j] = p;

        }
        else
        {
            nums2[j]++;
        }
    }


    for (i = 0; i<1000; i++)
    {
        if (word1[i]==word2[i])
        {
            if ('\0' == word1[i])
            {
                if ('\0' == word2[i])
                {
                    return true;
                }
                return false;
            }

            if (nums2[i] >= nums1[i])
            {
                continue;
            }
            else
            {
```

```
76                  return false;
77              }
78          }
79          return false;
80      }
81      return true;
82
83  }
```

执行结果： 通过 显示详情 >

执行用时： 0 ms ，在所有 C 提交中击败了 100.00% 的用户

内存消耗： 6.7 MB ，在所有 C 提交中击败了 98.45% 的用户