

#### 题目描述

给定一个字符串s和k，通过k次操作，需要将其转换为一个无重复的字符串。

- 如果前缀和后缀没有重复字符，则删除它们。
- 如果前缀和后缀没有重复字符，则删除它们。

约束：字符串s的长度在1到1000之间。

#### 输入描述

字符串s（长度在1到1000之间），k（操作次数）。

#### 输出描述

转换后的字符串。

#### 用例

输入	abcba
输出	abcba
说明	无
输入	abcbac
输出	abcbac
说明	无
输入	abcbac
输出	abcbac
说明	无
输入	.
输出	.
说明	无

#### 题目解析

给定字符串s和k，通过k次操作，需要将其转换为一个无重复的字符串。

如下操作：如果s的前缀和后缀没有重复字符，则删除它们。

因此，我们需要对字符串s进行递归操作，直到其变为一个无重复的字符串。

#### Java算法源码

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        String[] arr = s.split("");
        String prefix = arr[0];
        String suffix = arr[arr.length - 1];
        System.out.println(prefix + suffix);
    }

    public static String getResult(String prefix, String suffix) {
        String res = prefix + suffix;
        return res.replaceAll("(?>)", "");
    }
}
```

#### JavaScript源码

```
const readline = require('readline');

const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout,
});

rl.on('line', (line) => {
    const [prefix, suffix] = line.split(' ');
    let res = `${prefix}${suffix}`;
    res = res.replaceAll("(?>)", "");
    console.log(res);
});
```

在题目的用例中，有一个用例是“”，即空字符串。此时，我们需要对字符串s进行递归操作，直到其变为一个无重复的字符串。

### 3. Null 判断运算符

该运算符用于判断某个变量的值是否为null或undefined。如果某个变量的值为null或undefined，则返回true；否则返回false。

```
const headerText = response.settings.headerText || "Hello, world!";
const userAgent = response.settings.userAgent || "Mozilla/5.0";
const showSplashScreen = response.settings.showSplashScreen || true;
```

上面的三行代码都使用了||运算符进行默认赋值。但是，这样写是错误的。开发者的原意是，如果某个变量的值为null或undefined，则返回默认值。但是，如果某个变量的值为空字符串或false或0，则返回默认值。

为了避免这种情况，ES2020引入了一个新的Null判断运算符，它的行为类似于||，但只有当左侧的值为null或undefined时，才会返回右侧的值。

```
const headerText = response.settings.headerText ?? "Hello, world!";
const userAgent = response.settings.userAgent ?? "Mozilla/5.0";
const showSplashScreen = response.settings.showSplashScreen ?? true;
```

上面代码中，默认值只有在左侧属性值为null或undefined时，才会生效。

这里需要认识??和||的区别。

#### Python算法源码

```
import sys

s = input()
k = int(input())

def is_palindrome(s):
    return s == s[::-1]

def main():
    s = input()
    k = int(input())
    while k > 0:
        if is_palindrome(s):
            print(s)
            s = s[1:-1]
        else:
            print(s)
            s = s[1:-1]
        k -= 1

main()
```