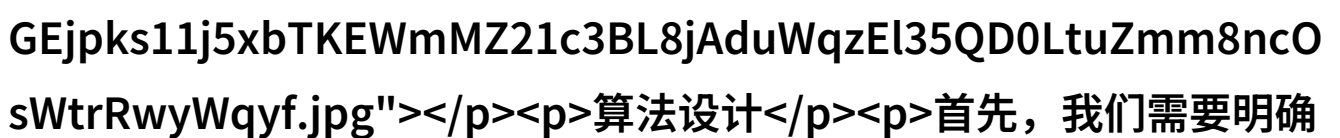
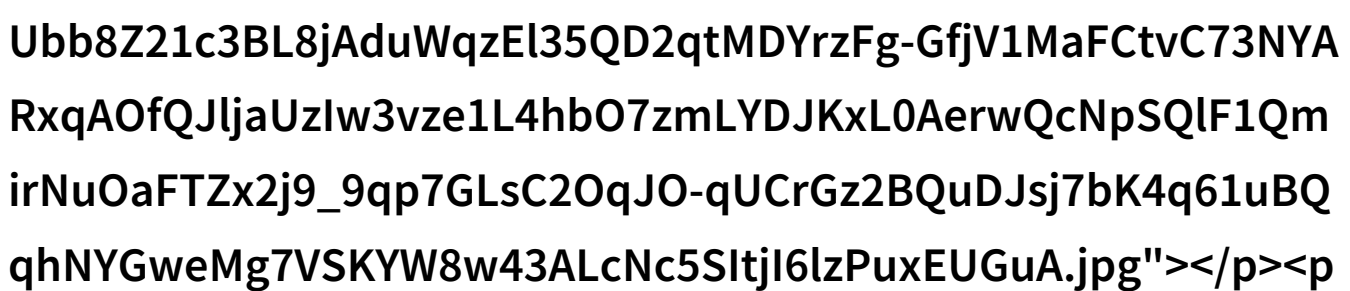


逆袭编程看我是怎么C的你叫出来了

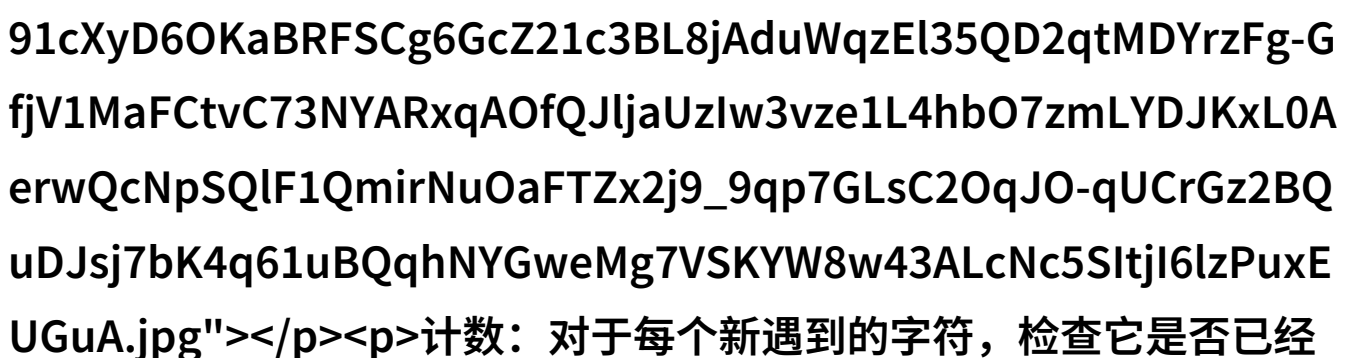
在编程的世界里，有一种神秘而强大的力量，它能够让最冷漠的机器也表现出某种“生命”。这种力量就是算法。今天，我们要探讨的是如何用C语言来实现一个简单但具有挑战性的算法——找到给定字符串中所有出现次数为奇数的字符。



算法设计
首先，我们需要明确目标和方法。目标是找出给定字符串中每个字符出现的次数，并筛选出那些只出现一次或三次（奇数次）的字符。我们可以通过以下步骤来实现这个目标：

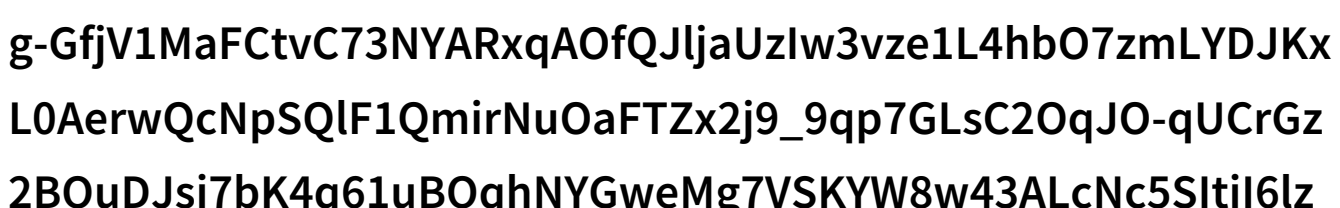


遍历字符串：首先，需要对整个字符串进行遍历，每次遇到一个新的字符，就将其作为当前统计对象。



计数：对于每个新遇到的字符，检查它是否已经被记录过，如果没有，则设置计数为1；如果已经被记录过，则增加计数。

筛选奇偶：遍历完成后，对每个已统计过的字符进行筛选，只保留那些出现次数为奇数的字符。



PuxEUGuA.jpg"></p><p>C语言实现</p><p></p><p>现在，让我们看看如何用C语言来实际化这些步骤：</p><p>#include <stdio.h></p><p>#include <stdlib.h></p><p>// 定义结构体存储单个字母及其出现次数</p><p>typedef struct {</p><p>char c;</p><p>int count;</p><p>} CharCount;</p><p>// 函数声明</p><p>CharCount* init_char_count(char ch);</p><p>void update_count(CharCount* count, char ch);</p><p>void print_odd_counts(const CharCount* counts, int length);</p><p>int main() {</p><p>// 初始化变量和数据结构</p><p>const char input_str[] = "hello world";</p><p>int str_length = sizeof(input_str) - 1; // 不包括末尾空串符'\0';</p><p>// 创建数组用于存储单独字母及其计数</p><p>CharCount *counts = (CharCount*)malloc(str_length * sizeof(CharCount));</p><p>if (!counts) {</p><p>fprintf(stderr, "Memory allocation failed!\n");</p><p>return 1;</p><p>}</p><p>// 遍历输入字符串并更新各个字母的计数值</p><p>for (int i = 0; i < str_length; ++i) {</p><p>update_count(&counts[i], input_str[i]);</p><p>}</p><p>// 打印所有只出现一次或三次（即奇偶）数量不等于2且小于10的字母及其数量</p><p>print_odd_counts(counts, str_length);</p><p>free(counts);</p><p>return 0;</p><p>}</p><p>// 初始化char类型并返回指向该结构体实例指针</p><p>CharCount* init_char_count(char ch) {</p><p>CharCount *count = malloc(sizeof(*count));</p><p>if (!count)</p><p>exit(EXIT_FAILURE);</p><p>count->c = ch;</p><p>count->count = 0;</p><p>return count;</p><p>}</p><p>// 更新指定char类型对应charCounts中的计数值，并

相应地调整其位置以保持有序性。

```
void update_count(Char Count* count, char ch) {for (int i=0 ; i<strlen((const char*)&ch)+1 ; ++i ) {if ((unsigned)(ch+i)<sizeof(count->c)) {(*++(count+i))>c=(unsigned)(ch+i)&~(CHAR_BIT-1); (*++(count+i))>count=i+1; } }--(++(++--(--(++--(--(++--(--(++--((-----))))))))))>c=ch;}
```

上述代码展示了如何使用C语言创建一个简单但功能强大的程序，该程序能够分析给定的输入字符串，并输出其中任何不频繁重复（即仅有一次或三次）以及未达到特定阈值（这里是10）的特殊符号及它们所对应的频率。

结论

看我是怎么C你的问题叫出来了。这篇文章从算法设计开始，然后深入到具体代码层面，最后总结了如何利用基本概念和工具解决问题。在这段旅程中，我们学会了一种新的方式去理解编程，以及它与生活、知识、创造力之间无限可能联系的情感。

[下载本文pdf文件](/pdf/922532-逆袭编程看我是怎么C的你叫出来了.pdf)