

C趣味程序百例(21)乘式还原 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022_C_E8_B6_A3_E5_91_B3_E7_A8_8B_c97_135445.htm 65.乘式还原(2) 有乘法

算式如下：
$$\begin{array}{r} \times \\ \hline \end{array}$$

----- 18个 的位置上全部是素数(1、3、5或7)，请还原此算式。 *问题分析与算法设计 问题中虽然有18数位，但只要确定乘数和被乘数后经过计算就可确定其它的数位。乘数和被乘数共有5个数位，要求每个数都是质数。完全可以采用穷举的方法对乘数和被乘数进行穷举，经过判断后找出答案。但是这种方法给人的感觉是“太笨了”，因为组成的数字只是质数(4个)，完全没有必要在那么大的范围内进行穷举，只需要试探每一位数字为质数时的情况即可。采用五重循环的方法实现对于5个数字的穷举，前面的许多例题中都已见过。循环实现简单易行，但嵌套的层次太多，需要穷举的变量的数量直接影响到循环嵌套的层数，这种简单的实现方法缺少技巧性。本例的程序中给出了另外一种同样功能的算法，该算法的实现思想请阅读程序。程序中并没有直接对质数进行穷举，而是将每个质数与1到4顺序一一对应，在穷举时为处理简单仅对1到4进行穷举处理，待要判断产生的乘积是否满足条件时再利用一个数组完成向对应质数的转换。请体会程序中的处理方法。程序中使用的算法实际上是回溯法。 *程序与程序注释 #include #define NUM 5 /*需要穷举的变量数目*/ #define C_NUM 4 /*每个变量的值的变化范围*/ int a[NUM 1]. /*为需要穷举的变量开辟的数组*/ /*a[1]:被乘数的百位，a[2]:十位,aa[3]:个位 a[4]:被乘数的十位 a[5]:个

```
位*/ int b[]={0,2,3,5,7}. /*存放质数数字的数组，不使用第0号  
元素*/ int f(long sum). void main() { int i,not_finish=1. i=2. /*i:将  
要进行处理的元素的指针下标。设置初始值*/ a[1]=1. /*为第1  
号元素设置初始值*/ while(not_finish) /*not_finish:程序运行没  
结束标记*/ { while(not_finish&i /*处理包括第i个元素在内的  
后续元素，找出当前条件下的一种各个变量的一种可能的取  
值方法*/ if(a[i]>=C_NUM) /*当要处理的元素取超过规定  
的C_NUM时*/ if(i==1&a[1]==C_NUM) not_finish=0. /*若1  
号元素已经到C_NUM，则处理全部结束*/ else a[i--]=0. /*将要  
处理的元素置0，下标-1(回退一个元素)*/ else a[i] . /*当前元素  
值加1后下标指针加1*/ if(not_finish) { 100Test 下载频道开通，  
各类考试题目直接下载。详细请访问 www.100test.com
```