

C趣味编程百例(08)自守数 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_BC_96_c97_135326.htm 27.自守数 自守数是指

一个数的平方的尾数等于该数自身的自然数。例如：25²=625

76²=5776 9376²=87909376 请求出200000以内的自守数 *题目分析

与算法设计 若采用“求出一个数的平方后再截取最后相应位数”的方法显然是不可取的，因为计算机无法表示过大的整数。

分析手工方式下整数平方(乘法)的计算过程，以376为例：

376 被乘数 X 376 乘数 ----- 2256 第一个部分积=被乘数

*乘数的倒数第一位 2632 第二个部分积=被乘数*乘数的倒

数第二位 1128 第三个部分积=被乘数*乘数的倒数第三位

----- 141376 积 本问题所关心的是积的最后三位。分析产生

积的后三位的过程，可以看出，在每一次的部分积中，并不是它的每一位都会对积的后三位产生影响。总结规律可以得到：

在三位数乘法中，对积的后三位产生影响的部分积分

别为：第一个部分积中：被乘数最后三位*乘数的倒数第一位

第二个部分积中：被乘数最后二位*乘数的倒数第二位 第三个

部分积中：被乘数最后一位*乘数的倒数第三位 将以上的部分

积的后三位求和后截取后三位就是三位数乘积的后三位。这样的规律可以推广到同样问题的不同位数乘积。按照手工计算的过程可以设计算法编写程序。 *程序说明与注释

```
#include
```

```
void main() { long mul,number,k,ll,kk. printf("It exists following
```

```
automorphic nmbers small than 200000:\n"). for(number=0.number
```

```
{ for(mul=number,k=1.(mul/=10)>0.k*=10). /*由number的位数
```

```
确定截取数字进行乘法时的系数k*/ kk=k*10. /*kk为截取部分
```

```
积的后三位求和后截取后三位就是三位数乘积的后三位。这样的规律可以推广到同样问题的不同位数乘积。按照手工计算的过程可以设计算法编写程序。 *程序说明与注释
```

```
#include
```

```
void main() { long mul,number,k,ll,kk. printf("It exists following
```

```
automorphic nmbers small than 200000:\n"). for(number=0.number
```

```
{ for(mul=number,k=1.(mul/=10)>0.k*=10). /*由number的位数
```

```
积时的系数*/ mul=0. /*积的最后n位*/ ll=10. /*ll为截取乘数相应位时的系数*/ while(k>0) { mul=(mul (number%(k*10))*(number%ll-number%(ll/10)))%kk. /*(部分积截取被乘数的后N位*截取乘数的第M位), %kk再截取部分积*/ k/=10. /*k为截取被乘数时的系数*/ ll*=10. } if(number==mul) /*判断若为自守数则输出*/ 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
```