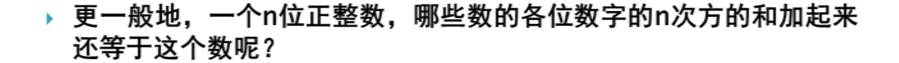
寻找自幂数

杨琦 西安交通大学计算机教学实验中心



【例】寻找自幂数

- 用户输入位数n,找出并显示出所有n位的自幂数。
- 数学家发现了很多有趣的数字。
- ▶ 比如, 153, 一个普通的三位数, 然而 $1^3 + 5^3 + 3^3 = 153$,
- 即它的各位数字的三次方的和等于这个数本身。



数学家称这样的数为自幂数,也叫自恋数。





n为1时, 自幂数称为独身, 0,1,2,3,4,5,6,7,8,9都是自幂数。

n为2时,没有自幂数。



n为3时, 自幂数称为水仙花数, 153就是一个水仙花数。

n=4, 称为四叶玫瑰数。

n=5, 称为五角星数。

n=6, 称为六合数。

n=7, 称为北斗七星数。

n=8, 称为八仙数。

n=9, 称为九九重阳数。



【问题分析】

- n位自幂数,各位数字的n次方的和加起来还等于这个数。
- ▶ 两个要点,一是怎样找出"各位",二是n次方的计算。
- ▶ n次方的计算有一个数学函数可用, pow(x,n)。
- 找出各位,举例来说,153,找个位,可用153%10=3;找十位呢? (153/10)%10=5,依次类推。直接求余,就是最低位的数字,除10,原来的十位就成为新的最低位,重复这一过程,就可以求出各位,直到这个数成为0。
- 还有一个问题要解决,就是构造n位数。0是最小的一位数,10的1次 方是最小的两位数,10的平方是最小的三位数,那么,10的n-1次 方就是最小的n位数。



【算法描述】

- ① 输入位数n。
- ②计算n位数的起始值和终止值 $start=10^{n-1}$, $end=10^{n-1}$, i=start
- ③ 如果i>end 转⑨。
- 4 m=i, sum=0
- ⑤ 如果m=0,转⑦。
- ⑥ d=m%10, sum=sum+dⁿ, m=m/10, 转⑤
- ⑦ 如果sum=i,显示i。
- ⑧ i=i+1, 转③
- ⑨ 结束。





【源程序】

//包含需要的头文件 #include <iostream> //数学函数需要的头文件 #include<cmath> //名字空间 using namespace std; int main() { //表示数的位数 int n; //表示n位数的起始值和终止值 int start, end; //待分解各位的数,即待判断的数 int m; int digit; //某个数位的值 //各位数的n次方的和 int sum; //循环变量, 待检验的数 int i;



```
cout<<"求n位自幂数,请输入位数:"; //提示信息
cin>>n; //输入位数
while(n>0){ //大于0时计算
              //n位数的起始值
 start=pow(10,n-1);
              //n位数的终止值
 end=pow(10,n)-1;
 cout<<n<<"位自幂数:"; //输出说明信息
 for(i=start;i<=end;i++){ //从起始值到终止值逐个检验
                   //将i赋给m
   m=i:
   //检验过程中m的值会改变,而i的值不变
   sum=0; //各位数的n次方和,检验前赋0
```



```
digit=m%10; //取最低位数字
sum=sum+pow(digit,n); //n次方, 再求和
m=m/10; //去掉个位, 刚才的十位成为新个位
//上面的循环结束时sum就是各位数字的n次方的和
if(sum==i){ //逻辑表达式的值为true时,表示是自幂数
  cout<<i<" "; //显示该数
```



```
    cout<<endl; //换行</li>
    cout<<"求n位自幂数,请输入位数:";</li>
    cin>>n; //再输入一个n表示位数
    /while循环
    cout<<endl;</li>
    return 0;
```





【运行结果】

求n位自幂数,请输入位数:1

1位自幂数:1 2 3 4 5 6 7 8 9

求n位自幂数,请输入位数:2

2位自幂数:

求n位自幂数,请输入位数:3

3位自幂数:153 370 371 407

求n位自幂数,请输入位数:4

4位自幂数:1634 8208 9474

求n位自幂数,请输入位数:5

5位自幂数:54748 92727 93084

求n位自幂数,请输入位数:0





【思路扩展】

- ①本例应掌握的技巧,
- 一是如何分离各位数字,



这实际是计算机科学中常用的一种"冗余"的思想,要获得某种保障,有意使用更多的时间、空间。





【思路扩展】

②分离各位数字的相反运算是合成一个数,例如有三个变量, a,b,c, 分别存放一位整数, 比如1,2,3, 如果将它们合成为a作百位, b作十位, c作各位的三位数呢?

③C++中, int型变量能表示的最大正整数为2147483647, 它不过10位, 那么有11,12,13位的自幂数吗?如果有,怎样计算呢?自幂数是有限的吗?如果有,有多少呢?

