**算法训练 寂寞的数**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　道德经曰：一生二，二生三，三生万物。  
　　对于任意正整数n，我们定义d(n)的值为为n加上组成n的各个数字的和。例如，d(23)=23+2+3=28, d(1481)=1481+1+4+8+1=1495。  
　　因此，给定了任意一个n作为起点，你可以构造如下一个递增序列：n,d(n),d(d(n)),d(d(d(n)))....例如，从33开始的递增序列为：  
　　33, 39, 51, 57, 69, 84, 96, 111, 114, 120, 123, 129, 141, ...  
　　我们把n叫做d(n)的生成元，在上面的数列中，33是39的生成元，39是51的生成元，等等。有一些数字甚至可以有两个生成元，比如101，可以由91和100生成。但也有一些数字没有任何生成元，如42。我们把这样的数字称为寂寞的数字。

输入格式

　　一行，一个正整数n。

输出格式

　　按照升序输出小于n的所有寂寞的数字，每行一个。

样例输入

40

样例输出

1  
3  
5  
7  
9  
20  
31

数据规模和约定

　　n<=10000

本题的C++参考代码如下：

#include <stdio.h>

int main()

{

int n;

int a[10000];

scanf("%d",&n);

for(int i=0;i<n;i++)

a[i]=0;

for(int i=1;i<n;i++)

{

int sum=0;

int j=i;

sum+=j;

while(j>0)

{

sum+=j%10;

j/=10;

}

if(sum<n) a[sum]=1;

}

for(int i=1;i<n;i++)

if(a[i]==0) printf("%d\n",i);

}

本题的C参考代码如下：

#include<stdlib.h>

#include<stdio.h>

int num(int n){

int m=n;

while(n!=0){

m+=n%10;

n/=10;

}

return m;

}

int main(){

int n,a[20000]={0},i; //考虑数组越界

scanf("%d",&n);

for(i=1;i<=n;i++){

a[num(i)]=1;

}

for(i=1;i<=n;i++){

if(a[i]==0){

printf("%d\n",i);

}

}

system("pause");

return 0;

}

本题的Java参考代码如下：

import java.util.\*;

public class Main {

public static void main(String args[])

{

Scanner read=new Scanner(System.in);

int n=read.nextInt();

int a[]=new int[n];

int b,temp;

for(int i=1;i<=n;i++){

b=i;

temp=i;

while(temp>0){

b+=temp%10;

temp=temp/10;

}

if(b<n)

a[b]++;

}

for(int i=0;i<n;i++){

if(a[i]==0&&i!=0)

System.out.println(i);

}

}

}