**回文数字**

问题描述

　　观察数字：12321，123321 都有一个共同的特征，无论从左到右读还是从右向左读，都是相同的。这样的数字叫做：回文数字。  
  
　　本题要求你找到一些5位或6位的十进制数字。满足如下要求：  
　　该数字的各个数位之和等于输入的整数。

输入格式

　　一个正整数 n (10<n<100), 表示要求满足的数位和。

输出格式

　　若干行，每行包含一个满足要求的5位或6位整数。  
　　数字按从小到大的顺序排列。  
　　如果没有满足条件的，输出：-1

样例输入

44

样例输出

99899  
499994  
589985  
598895  
679976  
688886  
697796  
769967  
778877  
787787  
796697  
859958  
868868  
877778  
886688  
895598  
949949  
958859  
967769  
976679  
985589  
994499

样例输入

60

样例输出

-1

本题的C++参考代码如下：

#include<stdio.h>

int main()

{

int n;

int flag=0;

scanf("%d",&n);

int i,j,k;

for(i=1;i<=9;i++)

for(j=0;j<=9;j++)

for(k=0;k<=9;k++)

{

if(2\*i+2\*j+k==n)

{

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

flag=1;

}

}

for(i=1;i<=9;i++)

for(j=0;j<=9;j++)

for(k=0;k<=9;k++)

{

if(2\*i+2\*j+2\*k==n)

{

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

flag=1;

}

}

if(flag==0)

printf("-1");

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

int main()

{

int a,b,c;

int n;

int flag=-1; //标志位

scanf("%d",&n);

for(a=1;a<10;a++)

{

for(b=0;b<10;b++)

{

for(c=0;c<10;c++)

{

if(a+b+c+b+a==n)

{

flag=1;

printf("%d%d%d%d%d\n",a,b,c,b,a);

}

}

}

}

for(a=1;a<10;a++)

{

for(b=0;b<10;b++)

{

for(c=0;c<10;c++)

{

if(a+b+c+c+b+a==n)

{

printf("%d%d%d%d%d%d\n",a,b,c,c,b,a);

flag=1;

}

}

}

}

if(flag==-1)

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

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main

{

public static void main(String args[]) throws IOException

{

BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

String str1 = bf.readLine();

int n = Integer.parseInt(str1);

int t = 0;

for (int i = 1; i <= 9; i++)

for (int j = 0; j <= 9; j++)

for (int k = 0; k <= 9; k++)

{

if (n == 2 \* (i + j) + k)

{

System.out.println(i \* 10000 + j \* 1000 + k \* 100 + j

\* 10 + i);

t = 1;

}

}

for (int i = 1; i <= 9; i++)

for (int j = 0; j <= 9; j++)

for (int k = 0; k <= 9; k++)

{

if (n == 2 \* (i + j + k))

{

System.out.println(i \* 100000 + j \* 10000 + k \* 1000

+ k \* 100 + j \* 10 + i);

t = 1;

}

}

if (t == 0)

System.out.println("-1");

}

}