**算法训练 数列**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　给定一个正整数k(3≤k≤15),把所有k的方幂及所有有限个互不相等的k的方幂之和构成一个递增的序列，例如，当k=3时，这个序列是：  
　　1，3，4，9，10，12，13，…  
　　（该序列实际上就是：30，31，30+31，32，30+32，31+32，30+31+32，…）  
　　请你求出这个序列的第N项的值（用10进制数表示）。  
　　例如，对于k=3，N=100，正确答案应该是981。

输入格式

　　只有1行，为2个正整数，用一个空格隔开：  
　　k N  
　　（k、N的含义与上述的问题描述一致，且3≤k≤15，10≤N≤1000）。

输出格式

　　计算结果，是一个正整数（在所有的测试数据中，结果均不超过2.1\*109）。（整数前不要有空格和其他符号）。

样例输入

3 100

样例输出

981

锦囊1

进制处理。

锦囊2

先把原数按二进制的方法转成二进制，然后再把转换后的数看成k进制，再转换回来。

本题的C++参考代码如下：

#include<iostream>

#include<cstdio>

#include<cstdlib>

using namespace std;

int main()

{

int k, n;

cin>>k>>n;

int a[1010]={0,1,k,1+k,k\*k};

for (int i=5; i<=n; i++)

if (i%2) a[i]=a[i-1]+1; else a[i]=a[2]\*a[i/2];

cout<<a[n];

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

int main()

{

int re[1000],k,n;

int p=0,t=1,i,j,f;

re[0]=1;j=0,f=1;

scanf("%d%d",&k,&n);

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

{

if(f==re[j])

{

f\*=k;

re[i]=f;

// printf("%d ",f);

j=0;

continue;

}

t=f+re[j++];

re[i]=t;

// printf("%d ",t);

}

printf("%d",re[i-1]);

return 0;

}

本题的Java参考代码如下：

import java.io.\*;

public class Main {

static long a[]=new long[1001];

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

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

String s[]=bf.readLine().split(" ");

int k=Integer.parseInt(s[0]);

int n=Integer.parseInt(s[1]);

int m=1;

int sum=0;

while(n>0){

a[m++]=n%2;

n=n/2;

}

m--;

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

sum+=a[i]\*add(k,i-1);

}

System.out.println(sum);

}

static int add(int k,int n){

int sum=1;

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

sum\*=k;

return sum;

}

}