**算法训练 进制转换**

时间限制：1.0s   内存限制：256.0MB

**问题描述**  
  
　　我们可以用这样的方式来表示一个十进制数： 将每个阿拉伯数字乘以一个以该数字所处位置的（值减１）为指数，以１０为底数的幂之和的形式。例如：１２３可表示为 １＊１０２＋２＊１０１＋３＊１００这样的形式。  
　　与之相似的，对二进制数来说，也可表示成每个二进制数码乘以一个以该数字所处位置的（值－１）为指数，以２为底数的幂之和的形式。一般说来，任何一个正整数Ｒ或一个负整数－Ｒ都可以被选来作为一个数制系统的基数。如果是以Ｒ或－Ｒ为基数，则需要用到的数码为 ０，１，．．．．Ｒ－１。例如，当Ｒ＝７时，所需用到的数码是０，１，２，３，４，５和６，这与其是Ｒ或－Ｒ无关。如果作为基数的数绝对值超过１０，则为了表示这些数码，通常使用英文字母来表示那些大于９的数码。例如对１６进制数来说，用Ａ表示１０，用Ｂ表示１１，用Ｃ表示１２，用Ｄ表示１３，用Ｅ表示１４，用Ｆ表示１５。  
　　在负进制数中是用－Ｒ 作为基数，例如－１５（十进制）相当于１１０００１（－２进制），并且它可以被表示为２的幂级数的和数：  
　　１１０００１＝１＊（－２）５＋１＊（－２）４＋０＊（－２）３＋０＊（－２）２＋  
　　０＊（－２）１ ＋１＊（－２）０  
　　 设计一个程序，读入一个十进制数和一个负进制数的基数, 并将此十进制数转换为此负进制下的数： －Ｒ∈｛－２，－３，－４，．．．，－２０｝  
  
**输入格式**  
　　一行两个数，第一个是十进制数Ｎ（－32768＜＝Ｎ＜＝32767）， 第二个是负进制数的基数－Ｒ。  
  
**输**出格式  
　　输出所求负进制数及其基数，若此基数超过１０，则参照１６进制的方式处理。（格式参照样例）  
  
　　**样例**输入1  
　　30000 -2

样例输出

30000=11011010101110000(base-2)

样例输入

-20000 -2

样例输出

-20000=1111011000100000(base-2)

样例输入

28800 -16

样例输出

28800=19180(base-16)

样例输入

-25000 -16

样例输出

-25000=7FB8(base-16)

锦囊1

使用进制转换的方法。

锦囊2

和普通的进制转换类似，还是除P取余倒着数，但注意取余是取正的部分。

本题的C++参考代码如下：

#include <iostream>

#include <cstdio>

using namespace std;

int N,M,base;

inline int Div(int a,int b)

{

int n;

n = a / b;

if(n\*b <= a)

return n;

return n + 1;

}

void work()

{

int num[100];

int top = 0;

if(N == 0)

{

cout << 0;

}

int P;

while(N)

{

P = Div(N,base);

num[++top] = N - P\*base;

N = P;

}

for(;top >= 1;top--)

{

if(num[top]<10)

cout << num[top];

if(num[top] >= 10)

cout << (char)(num[top] - 10 + 'A');

}

cout << "(base" << base << ")" << endl;

}

int main()

{;

while(cin >> N)

{

cin >> base;

M = N;

cout << N << "=";

work();

}

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

const char nc[20]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J'};

char str[20];

int main()

{

int m, n, k, t, s;

int i=0;

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

s = m;

while(m != 0)

{

k = m % n;

t = m / n;

if(k < 0)

{

k -= n;

t++;

}

m = t;

str[i++]=nc[k];

}

printf("%d=",s);

for(i = i- 1; i >= 0; i--)

printf("%c",str[i]);

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

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int N = scanner.nextInt();

int R = scanner.nextInt();

char[] c = "0123456789ABCDEFG".toCharArray();

String s1 = N + "=";

String s = "";

while (N != 0) {

int t = N % R;

if (t < 0) {

t = t - R;

N = N / R + 1;

} else

N = N / R;

s = c[t] + s;

}

System.out.println(s1 + s + "(base" + R + ")");

}

}