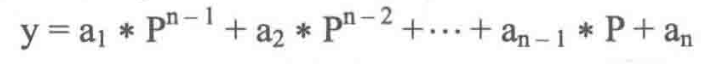
进制转换

十进制，二进制，八进制，十六进制之间如何转换？

如将一个p进制数x转换为十进制数y，则

设p进制数x为a1a2..an，那么转换成十进制数y就可以表示成：



从式子中可以看出，P的幂次从个位向高位是不断递增地，可以用循环来实现。从个位数开始乘以P的0次幂，之后用取余不断取出每个位子的数，P的幂次也随着位数不断递增，累加可以得到最终的十进制数。

//y是十进制数，product是P的累乘结果

Int y = 0, product = 1;

While(x != 0)

{  
 y = y + (x % 10) \* product; // x % 10取余获取x的个位数

x = x / 10; //整数相除去掉x的个位

product = product \* p;

}

将十进制数y转换为p进制数x

将y除以p，得到的余数作为低位存储在数组中，得到的商则继续除以p，直达商为0时，将得到的余数数组从高到低输出就是p进制数x。

如将7转换为二进制数：  
7 / 2 商为 3， 余数为1 低位

3 / 2 商为 1， 余数为1

1 / 2 商为 0， 余数为1 高位

所以，将余数的数组从高位到低位输出，即二进制数 111.

代码为：

Int x[20], num = 0; //余数数组x

Do

{  
 x[num++] = y % p; //y除以p取出余数

Y /= p; //得到商

}while(y != 0); //y是十进制数

数组x[num-1] 到 x[0]输出就是p进制数x。

实例 PAT B1022

题目：输入两个非负十进制整数A和B以及D（进制数），输出A+B的D（1 < D <= 10）进制数。

输入格式：

12 13 2

输出格式：  
11001

思路：

先相加，再用除以进制的方法求得余数。

代码如下：

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

using namespace std;

int main()

{

int a, b, d;

cin >> a >> b >> d;

//相加

int sum = a + b;

//设置存放转换后的数

int ans[31], num = 0;

do

{

ans[num++] = sum % d;

sum /= d;

} while (sum != 0);

//输出转换后的数

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

{

cout << ans[i];

}

return 0;

}