A number sequence is defined as follows:   
  
f(1) = 1, f(2) = 1, f(n) = (A \* f(n - 1) + B \* f(n - 2)) mod 7.   
  
Given A, B, and n, you are to calculate the value of f(n).

Input

The input consists of multiple test cases. Each test case contains 3 integers A, B and n on a single line (1 <= A, B <= 1000, 1 <= n <= 100,000,000). Three zeros signal the end of input and this test case is not to be processed.

Output

For each test case, print the value of f(n) on a single line.

Sample Input

1 1 3

1 2 10

0 0 0

Sample Output

2

5

又是一道给出了运算公式的数学，涛神还是没有说错，凡是没有优化的话，超时超内存等等是避免不了的了。这题很显然是一个找规律的题目，也就是该题的求解中是存在循环节的。

　　对于公式 f[n] = A \* f[n-1] + B \* f[n-2]; 后者只有7 \* 7 = 49 种可能，为什么这么说，因为对于f[n-1] 或者 f[n-2] 的取值只有 0,1,2,3,4,5,6 这7个数，A，B又是固定的，所以就只有49种可能值了。由该关系式得知每一项只与前两项发生关系，所以当连续的两项在前面出现过循环节出现了，注意循环节并不一定会是开始的 1，1 。 又因为一组测试数据中f[n]只有49中可能的答案，最坏的情况是所有的情况都遇到了，那么那也会在50次运算中产生循环节。找到循环节后，就可以轻松解决了。

#include <stdio.h>

#include <string.h>

int f[60];

int main()

{

int a,b,n,i;

f[0]=f[1]=1;

while(scanf("%d%d%d",&a,&b,&n))

{

if(a==0 &&b==0 &&n==0)return 0;

for(i=2;i<50;i++)

{

f[i]=(a\*f[i-1]+b\*f[i-2])%7;

if(f[i]==1 && f[i-1]==1)

break;

}

n%=(i-1);

if(n==0)

printf("%d\n",f[i-2]);

else

printf("%d\n",f[n-1]);

}

return 0;

}