**算法训练 Hanoi问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　如果将课本上的Hanoi塔问题稍做修改：仍然是给定N只盘子，3根柱子，但是允许每次最多移动相邻的M只盘子（当然移动盘子的数目也可以小于M）,最少需要多少次？  
　　例如N=5，M=2时，可以分别将最小的2个盘子、中间的2个盘子以及最大的一个盘子分别看作一个整体，这样可以转变为N=3，M=1的情况，共需要移动7次。

输入格式

　　输入数据仅有一行，包括两个数N和M（0<=M<=N<=8）

输出格式

　　仅输出一个数，表示需要移动的最少次数

样例输入

5 2

样例输出

7

本题的C++参考代码如下：

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int n,m;

cin>>n>>m;

if(n%m==0)

n/=m;

else n=n/m+1;

cout<<pow(2,n)-1;

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

int main()

{

int N,M;

scanf("%d%d",&N,&M);

N=(N+1)/M;

printf("%d",(1<<N)-1);

return 0;

}

本题的Java参考代码如下：

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int m=sc.nextInt();

int k=0;

if(n%m==0)

k=n/m;

else

k=(n+1)/m;

int count=1;

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

count\*=2;

System.out.println(count-1);

}

}