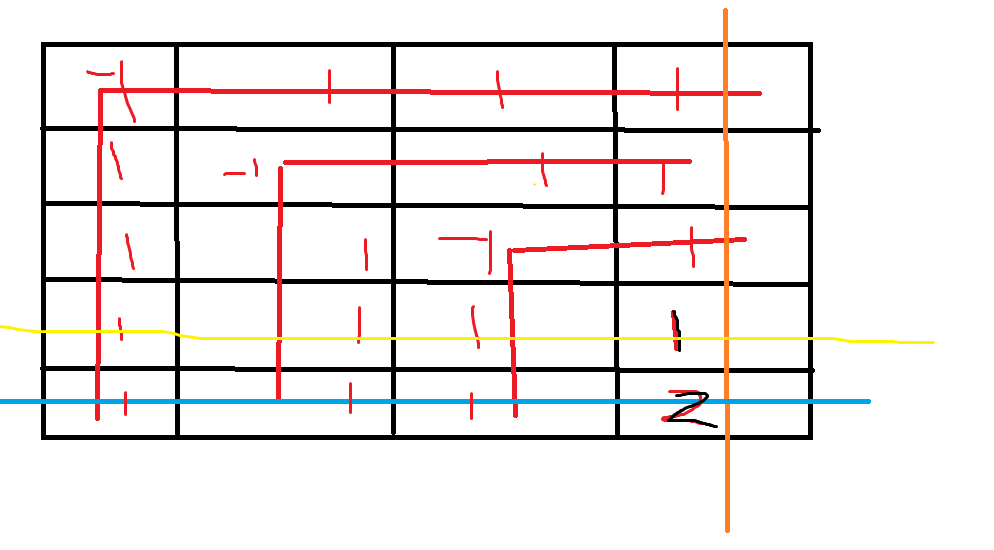
我们可以把最后一行和最后一列空出来，不管前面（n-1）\*（m-1）怎么填，我们都可以通过最后一行和最后一列的填数使得其满足要求

但只有一种情况，就是n和m奇偶性不同且k是-1的时候

红线相连的行列我们可以保证让他们两个满足要求而不影响其他的行列（总是将-1放在左上角）

这样，如果nm奇偶性不同，最后就会多出偶数个格子



那么我们还剩下橙线和蓝线和黄线的行列没处理

想让蓝线满足-1，格子2必须填-1,

想让黄线满足-1，格子1必须填-1

而这样的话，橙线却成了1，不满足题意了

如果k=1，那就好办了，全部填1都可以

所以当nm奇偶性不同且k=-1的时候，结果为0

其他的情况

2(n-1)(m-1)

需要用快速幂算

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <vector>

#include <algorithm>

#include<iostream>

using namespace std;

typedef long long ll;

const int mod=1000000007;

long long pow\_mod(long long a,long long n,long long m)//a^n mod m

{

long long res=1;

while(n>0)

{

if(n&1==1)

res=res\*a%m;

a=a\*a%m;

n>>=1;

}

return res;

}

//记得在最后输出结果的时候再模m一次

int main()

{

ll n,m,k;

cin>>n>>m>>k;

if(((n&1)==0 && (m&1)==1 || (n&1)==1 && (m&1)==0 ) && k==-1)

cout<<0<<endl;

else

{

ll ans=pow\_mod(pow\_mod(2,n-1,mod),m-1,mod);

cout<<ans<<endl;

}

return 0;

}