// 状态压缩 n\*m 中 填小方块的 方案数

#include <iostream>

#include <string.h>

#include <vector>

#define mod 1000000007

using namespace std;

int n,m;

int dp[21][1<<20];

vector<int> nexT[1<<20];

void dfs(int j,int i,int state,int nex)

{

if (j==m)

{

nexT[state].push\_back(nex);

// dp[i+1][nex]+=dp[i][state];

// dp[i+1][nex]%=mod;

return ;

}

//这个位置不能填

if (((1<<j)&state)>0) dfs(j+1,i,state,nex);

else

{

//竖着

dfs(j+1,i,state,nex|(1<<j));

//横着

if (j+1<m && ((1<<(j+1))&state)==0)

dfs(j+2,i,state,nex);

}

}

int main()

{

while (cin>>n>>m)

{

memset(dp,0,sizeof(dp));

for (int state=0;state<(1<<m);state++)

nexT[state].clear();

for (int state=0;state<(1<<m);state++)

dfs(0,0,state,0);

dp[0][0]=1;

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

for (int state=0;state<(1<<m);state++)

if (dp[i][state])

{

int len=nexT[state].size();

for (int j=0;j<len;j++)

{

int nex=nexT[state][j];

dp[i+1][nex]+=dp[i][state];

if (dp[i+1][nex]>=mod) dp[i+1][nex]-=mod;

}

}

cout<<dp[n][0]<<endl;

}

}