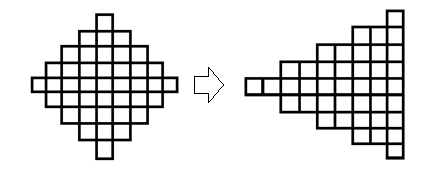
由于规模略大，所以考虑递推，但是首先要将菱形的图形转化为更加“规则”的图形才好找到递推方法。如图所示。



这样为递推提供了一个可行的顺序且不会影响最终结果。

设f[i][j]表示前i列里安置了j个妖兽且第i列安置了妖兽的方案数，设Len[i] 表示第i列有几行，那么有f[i][j]=∑f[k][j-1]\*(len[i]-(j-1))

因为前面已经选了j-1个，那么在第i列就有len[i]-（j-1）个位置可选。

为什么需要累加：例如对于前4排放置2个的情况(平移后的)，2个即可以放在第一列和第三列，也可以放在第一列和第四列，所以需要把这些分布在不同列的情况累加上来。

#include <bits/stdc++.h>

using namespace std;

int n,k,ans;

int len[210],f[210][210]= {1};

int main()

{

scanf("%d%d",&n,&k);

for(int i=1; i<=n; i++)

len[i\*2]=len[i\*2-1]=2\*i-1;

for(int i=1; i<=n\*2-1; i++)

for(int j=1; j<=i; j++)

for(int k=j-1; k<i; k++)

f[i][j]=(f[i][j]+f[k][j-1]\*(len[i]-j+1))%504;

for(int i=k; i<=n\*2-1; i++)

ans+=f[i][k];

printf("%d\n",ans%504);

return 0;

}