bool check(){

int cnt=1;

for(int i=1;i<N;i++){

int curr=row[i];

int prev=row[i-1];

//1. 같은 경우

if (curr==prev){

cnt++;

}

//2. 한 층 높은 경우

else if(curr-prev==1 ){

if (cnt>=L){ //경사로를 지을 수 있는 경우

cnt=1;

int j=i;

for(int j=1;j<=L;j++){

if (stair[i-j]==1) return false;

stair[i-j]=1;

}

}

//경사로를 지을 수 없는 경우 종료

else return false; }

//3. 한 층 낮은 경우

else if(curr-prev==-1){

int tmp=0;

//가능 한지 확인

for(int j=0;j+i<N;j++){

if (row[j+i]==curr){

tmp++;

}

else break;

}

//경사로 지을 수 있는 경우

if(tmp>=L){

cnt=1;

int j=i;

for(int j=0;j<L;j++){

if(stair[i+j]==1) return false;

stair[i+j]=1;

}

}

//경사로 지을 수 없는 경우 종료

else return false;

}

else{

return false;

}

}

return true;

}

int main(void){

cin>>N>>L;

int result=0;

//input

data=new int\*[N];

for (int i=0;i<N;i++){

data[i]=new int[N];

memset(data[i],0,sizeof(int)\*N);

}

row=new int[N];

stair=new int[N];

memset(row,0,sizeof(int)\*N);

memset(stair,0,sizeof(int)\*N);

for (int i=0;i<N;i++){

for (int j=0;j<N;j++){

cin>>data[i][j];

}

}

//rows

for (int i=0;i<N;i++){

for (int j=0;j<N;j++){

row[j]=data[i][j];

}

memset(stair,0,sizeof(int)\*N);

if (check()==true) {

result++;

}

}

//cols

for (int i=0;i<N;i++){

for (int j=0;j<N;j++){

row[j]=data[j][i];

}

memset(stair,0,sizeof(int)\*N);

if (check()==true) {

result++;

}

}

cout<<result;

return 0;

}