#include<bits/stdc++.h>

using namespace std;

vector<pair<int,int>>dir4={

    {0,1},{1,0},

    {0,-1},{-1,0}

};

vector<pair<int,int>>dir2={

    {0,1},{1,0}

};

int cnt = 0;

void Recrse(int i,int j,vector<vector<int>>&vec,vector<vector<bool>>&vis,int m,int n){

    if(i==m-1 && j==n-1){

        cnt++;

        return;

    }

    for(auto d:dir2){

        int nx = i+d.first;

        int ny = j+d.second;

        if(nx>=0 && nx<m && ny>=0 && ny<n && !vis[nx][ny] && vec[nx][ny]==1){

            vis[nx][ny]=true;

            Recrse(nx,ny,vec,vis,m,n);

            vis[nx][ny]=false;

        }

    }

}

int main(){

    int m,n,k;

    cin>>m>>n>>k;

    vector<vector<int>>vec(m,vector<int>(n,0));

    for(int i=0;i<m;i++){

        for(int j=0;j<n;j++){

            cin>>vec[i][j];

        }

    }

    vector<vector<bool>>vis(m,vector<bool>(n,false));

    queue<tuple<int,int,int>>q;

    for(int i=0;i<m;i++){

        for(int j=0;j<n;j++){

            if(vec[i][j]==1){

                q.push({i,j,0});

                vis[i][j]= true;

            }

        }

    }

    while(!q.empty()){

        auto it = q.front();

        q.pop();

        int x = get<0>(it);

        int y = get<1>(it);

        int d = get<2>(it);

        if(d==k){

            continue;

        }

        for(auto ti:dir4){

            int nx = x+ti.first;

            int ny = y+ti.second;

            if(nx>=0 && nx<m && ny>=0 && ny<n && !vis[nx][ny] && vec[nx][ny]==1){

                vis[nx][ny]=true;

                vec[nx][ny] = 0;

                q.push(make\_tuple(nx, ny, d + 1));

*// K--;*

            }

        }

    }

    Recrse(0,0,vec,vis,m,n);

    cout<<cnt<<endl;

    return 0;

}