

```

#include <bits/stdc++.h>

using namespace std;

vector<vector<int>>> graph;
vector<int> match;
vector<bool> visited;

int aug(int ini){
    if(visited[ini]) return 0;
    visited[ini]=1;
    for(int i=0;i<graph[ini].size();i++){
        int r=graph[ini][i];
        if(match[r]==-1 || aug(match[r])){
            match[r]=ini;
            return 1;
        }
    }
    return 0;
}

int main(){
    int r,c;
    string line;
    vector<string> listLines;
    vector<vector<pair<int,int>>>> cont;
    int cont1,cont2;
    vector<int> ver;
    int res;

    while(cin >> r >> c){
        for(int k=0;k<r;k++){
            cin>>line;
            listLines.push_back(line);
        }

        cont1 = cont2 = 0;

        cont.assign(r,vector<pair<int,int>>(c,pair<int,int>()));

        for(int i=0;i<r;i++){
            for(int j=0;j<c;j++){
                if((i+j)&1){
                    if(listLines[i][j]=='-'){
                        cont[i][j]=make_pair(cont1++,cont1);
                    }
                    else{
                        cont[i][j]=make_pair(cont1,-1);
                    }
                    cont1++;
                }
                else{
                    if(listLines[i][j]=='-'){
                        cont[i][j]=make_pair(cont2++,cont2);
                    }
                    else{
                        cont[i][j]=make_pair(cont2,-1);
                    }
                    cont2++;
                }
            }
            //cerr << i << " " << j << "->" << cont[i][j].first
            << " " << cont[i][j].second << endl;
        }

        for(int i=0;i<r;i++){
            for(int j=i%2;j<c;j+=2){
                if((i+1)<r){
                    ver.push_back(cont[i+1][j].first);
                }
            }
        }
    }
}

```

```

        if(cont[i+1][j].second != -1) ver.push_back(co
    nt[i+1][j].second);
    }
    if((j+1)<c){
        ver.push_back(cont[i][j+1].first);
        if(cont[i][j+1].second != -1) ver.push_back(co
    nt[i][j+1].second);
    }
    if((i-1)>=0){
        ver.push_back(cont[i-1][j].first);
        if(cont[i-1][j].second != -1) ver.push_back(co
    nt[i-1][j].second);
    }
    if((j-1)>=0){
        ver.push_back(cont[i][j-1].first);
        if(cont[i][j-1].second != -1) ver.push_back(co
    nt[i][j-1].second);
    }

    if(listLines[i][j]=='o'){
        graph.push_back(ver);
    }
    else{
        graph.push_back(ver);
        graph.push_back(ver);
    }

    //cerr << i << " " << j << "->";
    //for(int k=0;k<ver.size();k++) cerr << ver[k] << " "
;
    //cerr << endl;

    ver.clear();
}
}
match.assign(cont1,-1);
res=0;

for(int i=0;i<graph.size();i++){
    visited.assign(cont2,0);
    res+=aug(i);
}

if((cont1+cont2) == (2*res)) cout << "Y";
else cout << "N";
cout << endl;

graph.clear();
listLines.clear();

}
return 0;
}

```