#include<bits/stdc++.h>

using namespace std;

// sr - source row

// sc - source column

// dr - destination row

// dc - destination column

vector <string> getMazePaths( int dr, int dc) {

if(dr==0 || dc==0){

vector<string> z\_base;

return z\_base;

}

if(dr==1 && dc==1){

vector<string> base;

base.push\_back("");

return base;

}

vector<string> h\_zontal = getMazePaths(dr,dc-1);

vector<string> v\_zontal = getMazePaths(dr-1,dc);

vector<string> res;

for(string path:h\_zontal){

string str = "h"+path;

res.push\_back(str);

}

for(string path:v\_zontal){

string str = "v"+path;

res.push\_back(str);

}

return res;

}

vector<string> solve(int sr, int sc, int dr, int dc){

if(sr==dr && sc==dc){

vector<string> base;

base.push\_back("");

return base;

}

vector<string> hpaths;

vector<string> vpaths;

vector<string> res;

if(sr<dr){

hpaths = solve(sr+1,sc,dr,dc);

}

if(sc<dc){

vpaths = solve(sr,sc+1,dr,dc);

}

for(string path:hpaths){

string str = "h"+path;

res.push\_back(str);

}

for(string path:vpaths){

string str = "v"+path;

res.push\_back(str);

}

return res;

}

void display(vector<string>& arr){

cout << "[";

for(int i = 0;i < arr.size();i++){

cout << arr[i];

if(i < arr.size() -1) cout << ", ";

}

cout << "]"<<endl;

}

int main() {

int n,m; cin >> n >> m;

// vector<string> ans = getMazePaths(n,m);

vector<string> ans = solve(1,1,n,m);

display(ans);

return 0;

}