#include <iostream>

#include "ArgumentManager.h"

#include <fstream>

#include <vector>

using namespace std;

class player

{

int numships;

vector<vector<char>> board;

public:

player();

player(int inlength, int inwidth);

void setpos(int posx, int posy, int thing);

char getpos(int posx, int posy);

void shipincrement();

int getnumships();

void shipdecrement();

};

player::player()

{

numships = 0;

vector<vector<char>> createboard(10, vector<char>(10, NULL));

board = createboard;

}

player::player(int inlength, int inwidth)

{

numships = 0;

vector<vector<char>> createboard(inwidth, vector<char>(inlength, NULL));

board = createboard;

}

void player::setpos(int posx, int posy, int thing)

{

board[posx][posy]= thing;

}

char player::getpos(int posx, int posy)

{

return board[posx][posy];

}

void player::shipincrement() {

numships++;

}

void player::shipdecrement() {

numships--;

}

int player::getnumships() {

return numships;

}

int main(int argc, char\* argv[])

{

ArgumentManager am(argc, argv);

const string input = am.get("input");

const string output = am.get("output");

const string command = am.get("command");

ifstream inputFile;

ofstream outputFile;

inputFile.open(input);

outputFile.open(output);

int testcase, width, length, numshots, xshot, yshot;

int turn = 0;

char in;

vector<player> players(2);

inputFile >> testcase;

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

inputFile >> width >> length >> numshots;

player p1(length, width);

player p2(length, width);

players[0] = p1;

players[1] = p2;

for (int p = 0; p < 2; p++) {

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

for (int k = 0; k < length; k++) {

inputFile >> in;

players[p].setpos(j, k, in);

if (players[p].getpos(j, k) == '#') {

players[p].shipincrement();

}

}

}

}

bool end = false;

int turn = 0;

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

inputFile >> xshot >> yshot;

if (end) continue;

if (players[1 - turn].getpos(xshot, yshot) == '#') {

players[1 - turn].setpos(xshot, yshot, '\_');

players[1 - turn].shipdecrement();

if (players[1 - turn].getnumships() == 0) {

turn = 1 - turn;

if (turn == 0) {

end = true;

}

}

}

else {

turn = 1 - turn;

if (turn == 0 && players[0].getnumships() \* players[1].getnumships() == 0)

end = true;

}

}

if (players[0].getnumships() == 0 && players[1].getnumships() > 0) {

cout << "player one wins" << endl;

}

else if (players[0].getnumships() > 0 && players[1].getnumships() == 0) {

cout << "player two wins" << endl;

}

else {

cout << "draw"<<endl;

}

}

return 0;

}