Question 3(Kill enemies return Home)

\*Concept -> The concept of recursion is used to iterate through all possible paths to return home and kill soldiers if they come in path

\*Brief explanation of solution:-

* 1.A 2-d vector for representing the board
* 2.Taking coordinates of soldiers, Castle and Home to mark them as ‘S’ and ‘C’ and ‘H’ respectively.
* 3.Making a vector of pair to record path taken by castle.
* 4.Starting the recursion from castle’s position and moving to front every time.
* 5.When encountering a soldier, replacing ‘S’ with blank space and moving left
* 6.When reaching home coordinate stopping the recursion.
* 7.If coordinates of board is exceeded then stopping the recursion.

\*Code solution in cpp:-

#include<iostream>

#include<bits/stdc++.h>

using namespace std;

void find\_home\_path(vector<pair<int,int>>& answer, vector<vector<char>>& boards, pair<int,int>castle, pair<int,int> home)

{

if(castle.first<0 or castle.first>7 or castle.second<0 or castle.second>7)

{

    return;

}

if(castle.first==home.first and castle.second==home.second) //checking if current position is home

{

    answer.push\_back(make\_pair(castle.first,castle.second));

    return;

}

if(boards[castle.first][castle.second]=='S') //checking if current position has soldier

{

    boards[castle.first][castle.second]=' ';

    castle.first=castle.first-1;

    castle.second=castle.second;

    find\_home\_path(answer,boards,castle,home); //moving to the left after killing soldier

}

pair<int,int> o = make\_pair(castle.first,castle.second+1); //moving front normally

find\_home\_path(answer,boards,o,home); //moving to the front

}

int main(){

    vector<vector<char>>boards(8,vector<char>(8)); //making the board array

    int n;

    cout<<"Enter number of soldiers: "<<endl;

    cin>>n;

    for(int i=0;i<n;i++){

        int x,y;

        cout<<"Enter the position of soldier "<<i+1<<" (x,y): ";

        cin>>x>>y;

        if(x>7 and y>7 and x<0 and y<0){

            cout<<"Wrong coordinates";

            exit(0);

        }

        boards[x][y] = 'S';

}

cout<<"Enter position of castle: "<<endl;

int x,y;

cin>>x>>y;

boards[x][y] = 'C';

pair<int,int>castle\_position = make\_pair(x,y);

cout<<"Enter position of Home: "<<endl;

int x,y;

cin>>x>>y;

boards[x][y] = 'H';

pair<int,int>home= make\_pair(x,y);

vector<pair<int,int>> answer;

find\_home\_path(answer,boards,castle\_position,home);

}