**Project 2**

**<Clue V14>**

CIS-5 40561

Name: Jackson, Janaye

Date: 02/11/2022

# **Introduction**

Title: Clue V14

This is a computer version of Clue the board game.

Clue is played on a 24 by 23 cube board. A player’s position on the board is determined by the character they pick.

The game is set up by picking a random suspect, weapon, and room.

When the game begins, the player can role a set of dice or make an accusation.

If the player decides to make an accusation, they pick who they believe did it, the weapon used, and where it happened. If they guess correctly, they win. If they guess wrong their turn starts over.

If the player decides to roll the dice, the player can move up, down, left or right a number of times set by a role of the dice. If they end up in a room they can make an accusation, name a suspect, or move on to their next turn.

If the player makes an accusation in a room, the same rules as stated previously apply except the room is where the player is located.

If the player names a suspect, they are told whether their suspicion confirmed or denied.

After each turn, they can bring up a list of all of their suspects.

# **Summary**

Project size: about 630 lines

Comments: about 140 lines

The number of variables: about 35

This project includes only the concepts that have been learned from the chapters in the book dedicated to this project. I have been able to simplify a lot of this project into different functions, but I still do not believe it is as efficient as it could be. With more education I believe I can further improve the game.

There are a numerous amount of things I still do not know, so the possibilities are endless.

It took about a week to do, I wanted to follow the same pattern as my previous project. I took quite a few breaks and sometimes just spontaneously started working on it. I come up with a lot of new ideas when I am left to think a lone. I could then come back and see my code in a new perspective. I did not have too much trouble; My final product is still long, but I am glad the main function is not the entirety of the code.

I am mostly satisfied with my project and look forward to improving it.

# **Description**

The main point of this project is to replicate a board game.

## **Flowchart**

Diagram

Description automatically generated

I know the flowchart may be hard to read, so I included a downloadable pdf

File: 

## **Pseudocode**

**Main**

Initialize random seed

Declare Variables

Initialize Variables

setCha()

setWpon()

setRoom()

Initialize Variables

Create output and input variables

Do

Display Output

Print()

Display Output

User Input

Initialize Variable

setLoc=()

Display Output

User Input

While(ch = n or N)

Initialize Variables

Open file

Do

Display Output

locRoom()

Display Output

User Input

If (ch = y or Y)

(Display Output

Print()) \*3

Display Output

User Input

If(true && true && true)

Initialize Variable

Else

Initialize Variable

Output to file

Else

Display Output

Initialize Variable

Display Output

For Loop

Display Output

LocRoom()

Display output

User Input

If(ch = u or U)

Increment Variable

If(vPos > 24)

Display Output

Decrement Variables

Else If(ch = d or D)

Decrement Variable

If(vPos < 0)

Display Output

Increment Variables

Else If(ch = l or L)

Decrement Variable

If(hPos < 0)

Display Output

Increment Variables

Else If(ch = r or R)

Increment Variable

If(hPos > 23)

Display Output

Decrement Variables

Else if(ch = n or N)

Initialize Varibale

Else

Display output

Increment Variable

Check()

Initialize Variable

If(hint = 70)

Display Output

Initialize Variable

If(slct1 = pick)

Increment Variable

Display Output

Initialize Variable

If(inRoom = true)

Display Output

User Input

If(ch = y or Y)

(Display Output

Print())\*3

If(true && true && true)

Initialize Variable

Else

Display Output

Output to File

Else

Display Output

User Input

If(ch = y or Y)

(Display Output

Print())\*3

Display Output

User Input

If(choice < 6)

If(choice != slct1)

Display Output

Initialize Variable

Pects()

Else

Display Output

Initialize Variable

Pects()

If(choice >= 6 && choice <12)

If(choice != slct2)

Display Output

Initialize Variable

Pects()

Else

Display Output

Initialize Variable

Pects()

If(choice >= 12 && choice < 21)

If(choice != slct3)

Display Output

Initialize Variable

Pects()

Else

Display Output

Initialize Variable

Pects()

Else

Display Output

Suspect()

Display Output

User Input

If(ch == y or Y)

Open file

readIn()

Close file

Display Output

User Input

If(ch = y or Y)

Initialize Variable

Print()

While(win == false)

Close file

Display Output

Return 0

**Check()**

Initialize Variable

If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else If(true and true)

Initialize Variable

Else

Initialize Variable

Return inRoom

**Accus()**

Initialize Variable

User Input

If(ac1 != slc1)

Initialize Variable

Else if(ac2 != slc2)

Initialize Variable

Else if(ac3 != slct3)

Initialize Variable

Return match

**linSrch()**

forLoop

if(inp[i] = look)

return i

return -1

**setLoc()**

LinSrch()

If(check < 0)

Exit(1)

Display Output

Initialize Variables

Return hPos

**Pects()**

Initialize Variables

Return num

**locRoom()**

Display Output

**Print()**

Switch(cha)

Case a

Forloop

Display Output

Case b

Forloop

Display Output

Case c

Forloop

Display Output

Case d

Forloop

Display Output

**Print()**

Display Output

Forloop

Display Output

Forloop

Display Output

Forloop

Display Output

Forloop

Display Output

**selSort()**

forloop

Initialize Variables

For loop

If(inp[n] < temp)

Initialize Variables

Initialize Variables

**bubSort()**

Declare Variables

Do

Initialize Variables

Forloop

If(inp[i] > inp[i+1]

Initialize variables

While(swap)

**setCha()**

Initialize Variables

**setWpon()**

Initialize Variables

**setRoom()**

Initialize Variables

**Suspect()**

If(cha.size() >1)

selSort()

If(wpon.size() >1)

selSort()

If(room.size() >1)

bubSort()

If(guilt.size()>1)

bubSort()

forLoop

Initialize Variable

forLoop

Initialize Variable

forLoop

Initialize Variable

forLoop

Initialize Variable

**readIn()**

Initialize variable

While(in>>spect)

If(spect = lEnd)

Display Output

Else

Display Output

**readTo()**

Output to File

## **Major Variables**

|  |  |  |
| --- | --- | --- |
| **Type** | **Variable Name** | **Description** |
| **String** | file | Constant string for file name |
|  | killer | String for killer |
|  | wpon | String for weapon used |
|  | room | String for room used |
|  | pRoom | String for character location |
|  | spect | String for suspect |
|  | inCha | String for setting character |
|  | in | String for setting character |
|  |  |  |
| **String Arrays** | cha | String for all playable characters |
|  | wpons | String for all weapons |
|  | rooms | String for all rooms |
|  | susp | 2 dimensional array for all suspicion |
|  |  |  |
| **String Vectors** | kill | Vector for suspected killers |
|  | wpn | Vector for suspected weapons |
|  | rm | Vector for suspected rooms |
|  | guilty | Vector for confirmed suspicions |
|  |  |  |
| **Integer** | choice | Integer for selections |
|  | dice | Integer for dice |
|  | slct1-3 | Integer for case selections |
|  | ac1-3 | Integer for user accusations |
|  | sus | Integer for user suspects |
|  | vPos | Integer for user position |
|  | hPos | Integer for user position |
|  | pos | Integer for user position |
|  | pick | Integer for hint |
|  |  |  |
| **Integer Array** | v | Integer for all room locations – vertical |
|  | h | Integer for all room locations – horizontal |
|  |  |  |
| **Float** | hint | Float for hint |
|  |  |  |
| **Character** | ch | Character for user input |
|  |  |  |
| **Boolean** | win | Boolean for win |
|  | inRoom | Boolean for character in room |

## **C++ Constructs**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Section** | **Topic** |
| 2 | 2 | cout |
|  | 3 | libraries |
|  | 4 | variables/literals |
|  | 5 | Identifiers |
|  | 6 | Integers |
|  | 7 | Characters |
|  | 8 | Strings |
|  | 9 | Floats No Doubles |
|  | 10 | Bools |
|  | 12 | Variables 7 characters or less |
|  | 14 | Arithmetic operators |
|  | 15 | Comments 20%+ |
|  | 16 | Named Constants |
|  |  |  |
| 3 | 1 | cin |
|  | 2 | Math Expression |
|  | 5 | Type Casting |
|  | 7 | Formatting output |
|  | 8 | Strings |
|  | 9 | Math Library |
|  |  |  |
| 4 | 1 | Relational Operators |
|  | 2 | if |
|  | 4 | If-else |
|  | 5 | Nesting |
|  | 6 | If-else-if |
|  | 8 | Logical operators |
|  | 11 | Validating user input |
|  | 13 | Conditional Operator |
|  | 14 | Switch |
|  |  |  |
| 5 | 1 | Increment/Decrement |
|  | 2 | While |
|  | 5 | Do-while |
|  | 6 | For loop |
|  | 11 | Files input/output both |
|  |  |  |
| 6 |  | Functions |
|  | 3 | Function Prototypes |
|  | 5 | Pass by Value |
|  | 8 | Return |
|  | 9 | Returning Boolean |
|  | 11 | Static Variables |
|  | 12 | Defaulted arguments |
|  | 13 | Pass by reference |
|  | 14 | Overloading |
|  | 15 | Exit() function |
|  |  |  |
| 7 |  | Arrays |
|  | 1 to 6 | Single Dimensioned Arrays |
|  | 7 | Parallel Arrays |
|  | 8 | Single Dimensioned as Function Arguments |
|  | 9 | 2 Dimensioned Arrays |
|  | 12 | STL Vectors |
|  |  | Passing Arrays to and from Functions |
|  |  | Passing Values to and from Functions |
|  |  |  |
| 8 |  | Searching and Sorting Arrays |
|  | 3 | Bubble Sort |
|  | 3 | Selection Sort |
|  | 1 | Linear Search |

## **Reference**

1. Problem Solving with C++ - Walter Savitch
2. Starting out with C++ - Tony Gaddis
3. Class Lectures

## **Program**

/\*

\*

\* File: main.cpp

\* Author: Janaye Jackson

\*

\* Created on February 10, 2022, 6:00 PM

\* Purpose: Clue V14

\*/

//System Level Libraries

#include <cmath> //Math Library

#include <cstdlib> //Random number Library

#include <ctime> //Time to set random seed

#include <fstream> //FileLibrary

#include <iostream> //Input-Output Library

#include <iomanip> //Format Library

#include <vector> //Vectors Library

using namespace std;

//User Defined Libraries

//Global Constants, not Global Variables

//These are recognized constants from the sciences

//Physics/Chemistry/Engineering and Conversions between

//systems of units

//Function Prototypes

bool accus(int &, int&, int&, int, int, int); //Check to see if user accusation is false

bool check(int, int, string &, string []); //Check if user is in a room

int pects(); //Number of suspects character named

int setLoc(string, string[], int, int &, int [], int []); //Set user position vertical and horizontal

void linSrch(string, string[]); //Linear Search

void locRoom(); //Display position of rooms

void print(string [], int, char); //Print string arrays

void print(string [][4], int = 0, int = 0, int = 0, int = 0); //Print 2 dimensional arrays

void selSort(vector<string> &); //Selection Sort

void bubSort(vector<string> &); //Bubble Sort

void setCha(string[]); //Set characters in an array

void setWpon(string[]); //Set weapons in an array

void setRoom(string[]); //Set rooms in an array

void suspect(vector<string>, vector<string>, vector<string>, vector<string>, string [][4]); //Set 2 dimensional array with vectors

void readIn(fstream &, string &); //Read in from a file

void readTo(fstream &, string, string, string); //Output to a file

//Execution begins here!

int main(int argc, char\*\* argv) {

//Initialize Random Seed once here!

srand(static\_cast<unsigned int>(time(0)));

//Local Constants

const string file = "SuspectList.txt"; //String for file names

const int size1 = 6; //Integer for size of array

const int size2 = 9; //Integer for size of array

//String arrays

string cha[size1]; //Strings for all the playable characters

string wpons[size1]; //Strings for all of the weapons

string rooms[size2]; //Strings for all of the rooms

string susp[15][4]; //2 Dimensional array for suspects

//String vectors

vector<string> kill, wpn, rm, guilty; //String vectors for user suspicion

//Strings

string killer, wpon, room; //Strings for the killer, the weapon used, and room used

string pRoom, spect; //String for room player is in and who they suspect

string inCha, in; //String for setting character

//Integer arrays

int v[size1] = {17, 24, 19, 0, 0, 6}; //Integer array for character positions

int h[size1] = {23, 16, 0, 9, 14, 0};

//Integers

int choice; //User input for player to pick suspect and character

int dice; //Dice

int slct1, slct2, slct3; //Integers for case file picks

int ac1, ac2, ac3, sus; //Integers for user accusations or suspects

int vPos, hPos; //Integers for player position

int pick; //Integer for player hint

int sPct; //Integer for suspects user named

//Floats

float hint; //Float for player hint

//Chars

char ch; //User input for player to answer questions

//Booleans

bool win, inRoom; //Booleans for player win and for if player is in a room

//Set strings to name of players

setCha(cha);

//Set strings to name of weapons

setWpon(wpons);

//Set strings to name of rooms

setRoom(rooms);

//Initialize Integers

sPct = 0;

//Initialize boolean

win = false;

//fstream

fstream out;

fstream input;

//Player selection

do{

//List of players

cout<<" Players List"<<endl;

print(cha, size1, 'a');

cout<<"\nWhich player would you like to play as? (Type in the character name)"<<endl; //User choice

cin>>inCha>>in;

cout<<endl;

inCha += " "+in;

hPos = setLoc(inCha, cha, size1, vPos, v, h);

//Character conformation

cout<<"Is this the character you want? (y or n)"<<endl;

cin>>ch;

}while(ch == 'n' || ch == 'N');

//Case File - Who, with what, where

slct1 = rand()%6;

killer = cha[slct1];

slct2 = rand()%6;

wpon = wpons[slct2];

slct3 = rand()%9;

room = rooms[slct3];

//Open file to place suspect

out.open(file, ios:: out);

//Game play

do{

//output player location

cout<<"\nYou are at v-"<<vPos<<" h-"<<hPos<<endl;

//Output room locations

locRoom();

//Player choice

cout<<"Would you like to make an accusation? (y or n)"<<endl;

cin>>ch;

//Provide accusation choices

if(ch == 'y' || ch == 'Y'){

//List players to choose from

cout<<"\n Players List"<<endl;

print(cha, size1, 'b');

//List weapons to choose from

cout<<"\n Weapons List"<<endl;

print(wpons, size1, 'b');

//List rooms to choose from

cout<<"\n Rooms List"<<endl;

print(rooms, size2, 'b');

//User accusation

cout<<"Choose a person, a weapon, and a room - (Ex. 1 1 1 would pick "<<cha[0]<<" & "<<wpons[0]<<" & "<<rooms[0]<<")"<<endl;

cin>>ac1>>ac2>>ac3;

if(ac1 == slct1 && ac2 == slct2 && ac3 == slct3)

{

win = true; //if all accusations are true player wins

}

else{

cout<<"\nYour accusation is incorrect"<<endl;

readTo(out, cha[ac1], wpons[ac2], rooms[ac3]);

}

}

//Else continue game

else{

//Dice rolling

cout<<"\nRolling Dice..."<<endl;

dice = rand()%12+1;

cout<<"You rolled a "<<dice<<endl;

//User movement

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

//Output player location

cout<<"Your position is v-"<<vPos<<", h-"<<hPos<<endl;

//Output room locations

locRoom();

//Player movement

cout<<"\nYou have "<<dice-i<<" spaces left to move."<<endl;

cout<<"Would you like to move up, down, left, right, or not at all? (u, d, l, r, n)"<<endl;

cin>>ch;

if(ch == 'u' || ch == 'U'){ //Moving up

vPos++;

if(vPos > 24){

cout<<"You cannot move that way"<<endl;

vPos--;

i--;

}

}

else if(ch == 'd' || ch == 'D'){ //Moving down

vPos--;

if(vPos < 0){

cout<<"You cannot move that way"<<endl;

vPos++;

i--;

}

}

else if(ch == 'l' || ch == 'L'){ //Moving to the left

hPos--;

if(hPos < 0){

cout<<"You cannot move that way"<<endl;

hPos++;

i--;

}

}

else if(ch == 'r' || ch == 'R'){ //Moving to the right

hPos++;

if(hPos > 23){

cout<<"You cannot move that way"<<endl;

hPos--;

i--;

}

}

else if(ch == 'n' || ch == 'N'){ //Not moving

i = dice;

}

else{

cout<<"Invalid Option"<<endl;

i--;

}

}

//Is character in a room

inRoom = check(vPos, hPos, pRoom, rooms);

//Initialize hint

hint = sqrt(pow(vPos, 3));

//Giving player semi-random hint

if(static\_cast<int>(hint) == 70){

cout<<"You have received a hint:"<<endl;

//Initialize hint

pick = (dice-1) / 2;

//Pick hint to give

if(slct1 == pick){

pick++;

}

spect = cha[pick];

cout<<spect<<" is not a suspect."<<endl;

kill.push\_back(spect);

}

if(inRoom == true){ //Player is in room

cout<<"\nYou are in "<<pRoom<<endl;

//Does user know who, what, and where?

cout<<"Would you like to make an accusation?(y or n)"<<endl;

cin>>ch;

if(ch == 'y' || ch == 'Y')//Provide Accusation choices

{

//List players to choose from

cout<<"\n Players List"<<endl;

print(cha, size1, 'b');

//List weapons to choose from

cout<<"\n Weapons List"<<endl;

print(wpons, size1, 'b');

//User accusation

cout<<"Choose a person and a weapon - (Ex. 1 1 would pick "<<cha[0]<<" & "<<wpons[0]<<")"<<endl;

cin>>ac1>>ac2;

if(ac1 == slct1 && ac2 == slct2 && pRoom == room)

{

win = true; //If all accusation are true player win

}

else

{

cout<<"Your accusation is incorrect"<<endl;

readTo(out, cha[ac1], wpons[ac2], pRoom);

}

}

else{

//Does user suspect and item or person

cout<<"Would you like to name a suspect? (y or n)"<<endl;

cin>>ch;

if(ch == 'y' || ch == 'Y'){ //Provide list of suspect

//List players to choose from

cout<<"\n Players List"<<endl;

print(cha, size1, 'b');

//List weapons to choose from

cout<<"\n Weapons List"<<endl;

print(wpons, size1, 'c');

//List rooms to choose from

cout<<"\n Rooms List"<<endl;

print(rooms, size2, 'd');

cout<<"Choose who or what you suspect - (Ex. 1 would pick "<<cha[0]<<")"<<endl;

cin>>choice;

}

//Determine is user suspicion correct

if(choice < 6){ //Character suspect

if(choice != slct1){

cout<<"This is not a suspect. A user has this character."<<endl;

//Add Suspect to vector

kill.push\_back(cha[choice]);

sPct = pects();

}

else{

cout<<"No user has this character. This is a suspect."<<endl;

//Add killer to vector

guilty.push\_back(cha[choice]);

sPct = pects();

}

}

else if(choice >= 6 && choice < 12){ //Weapon suspect

if(choice - 6 != slct2){

cout<<"This is not the weapon used. A user has this weapon."<<endl;

//Add weapon to vector

wpn.push\_back(wpons[choice - 6]);

sPct = pects();

}

else{

cout<<"No user has this weapon. This is a weapon used."<<endl;

//Add weapon to vector

guilty.push\_back(wpons[choice - 6]);

sPct = pects();

}

}

else if(choice >=12 && choice <21){ //Room suspect

if(choice - 12 != slct3){

cout<<"This is not where the murder happened. A user has this room."<<endl;

//Add room to vector

rm.push\_back(rooms[choice - 12]);

sPct = pects();

}

else{

cout<<"No user has this room. This is where the murder happened."<<endl;

//Add room to vector

guilty.push\_back(rooms[choice - 12]);

sPct = pects();

}

}

}

}

else

{

cout<<"You are not in any room"<<endl; //User is not in a room

}

suspect(kill, wpn, rm, guilty, susp);

cout<<"\nWould you like to see your accusation list? (y or n)"<<endl;

cin>>ch;

if(ch == 'y' || ch == 'Y'){

//Open file

input.open(file.c\_str(), ios::in);

readIn(input, spect);

//Close file

input.close();

}

cout<<"\nYou have "<<sPct<<" suspect(s)";

cout<<"\nWould you like to see your suspect list? (y or n)"<<endl;

cin>>ch;

if(ch == 'y' || ch == 'Y'){

int a = kill.size();

int b = wpn.size();

int c = rm.size();

int d = guilty.size();

print(susp, a, b, c, d);

}

}

}while(win == false); //Continue until player wins

//Close file

out.close();

//Display outputs

cout<<"You won!"<<endl;

cout<<"The killer was "<<killer<<" the weapon was "<<wpon<<" and it happened in the "<<room<<" room."<<endl;

//Exit the program

return 0;

}

bool check(int vPos, int hPos, string &pRoom, string inp[]){

bool inRoom;

if((vPos>=0 && vPos<=5) && (hPos>=0 && hPos<=5)){ //Conservatory

inRoom = true;

pRoom = inp[0];

}

else if((vPos>= 8 && vPos<=12) && (hPos>=0 && hPos<=5)){ //Billiard Room

inRoom = true;

pRoom = inp[1];

}

else if((vPos>=14 && vPos<=18)&& (hPos>=0 && hPos<=6)){ //Library

inRoom = true;

pRoom = inp[2];

}

else if((vPos>=21 && vPos<=24) && (hPos>=0 && hPos<=6)){ //Study

inRoom = true;

pRoom = inp[3];

}

else if((vPos>=0 && vPos<=7) && (hPos>=8 && hPos<=15)){ //Ballroom

inRoom = true;

pRoom = inp[4];

}

else if((vPos>=18 && vPos<=24) && (hPos>=9 && hPos<=14)){ //Hall

inRoom = true;

pRoom = inp[5];

}

else if((vPos>=0 && vPos<= 6) && (hPos>=18 && hPos<=23)){ //Kitchen

inRoom = true;

pRoom = inp[6];

}

else if((vPos>=9 && vPos<=15) && (hPos>=16 && hPos<=23)){ //Dining Room

inRoom = true;

pRoom = inp[7];

}

else if((vPos>=19 && vPos<=24) && (hPos>=17 && hPos<=23)){ //Lounge

inRoom = true;

pRoom = inp[8];

}

else{

inRoom = false;

}

return inRoom;

}

bool accus(int &ac1, int &ac2, int &ac3, int slc1, int slc2, int slc3){

bool match = true;

cin>>ac1>>ac2>>ac3;

if(ac1 != slc1)

match = false;

else if(ac2 != slc2)

match = false;

else if(ac3 != slc3)

match = false;

return match;

}

int linSrch(string look, string inp[], int size){

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

if(inp[i] == look)

return i;

}

return -1;

}

int setLoc(string ch, string cha[], int size, int &vPos, int v[], int h[]){

int check = linSrch(ch, cha, size);

if(check < 0)

exit(1);

cout<<"You are "<<cha[check]<<endl;

vPos = v[check];

int hPos = h[check];

return hPos;

}

int pects(){

static int num;

num++;

return num;

}

void locRoom(){

cout<<"\nRoom | Vertical Location | Horizontal Location |"<<endl;

cout<<"Conservatory | between 0 and 5 | between 0 and 5 |"<<endl;

cout<<"Billiard Room | between 8 and 12 | between 0 and 5 |"<<endl;

cout<<"Library | between 14 and 18 | between 0 and 6 |"<<endl;

cout<<"Study | between 21 and 24 | between 0 and 6 |"<<endl;

cout<<"Ballroom | between 0 and 7 | between 8 and 15 |"<<endl;

cout<<"Hall | between 18 and 24 | between 9 and 14 |"<<endl;

cout<<"Kitchen | between 0 and 6 | between 18 and 23 |"<<endl;

cout<<"Dining Room | between 9 and 15 | between 16 and 23 |"<<endl;

cout<<"Lounge | between 19 and 24 | between 17 and 23 |"<<endl;

}

void print(string inp[], int size, char cha){

switch(cha){

case 'a': {

for(int i = 0; i < size; i++)

cout<<inp[i]<<endl;

break;

}

case 'b':{

for(int i = 0; i < size; i++)

cout<<i<<" - "<<setw(15)<<inp[i]<<endl;

break;

}

case 'c':{

for(int i = 0; i < size; i++)

cout<<i + 6<<" - "<<setw(15)<<inp[i]<<endl;

break;

}

case 'd':{

for(int i = 0; i < size; i++)

cout<<i+12<<" - "<<setw(15)<<inp[i]<<endl;

break;

}

}

}

void print(string inp[][4], int a, int b, int c, int d){

cout<<"Suspect - ";

for(int i = 0; i < a; i++)

cout<<inp[i][0]<<", ";

cout<<endl<<"Weapon - ";

for(int i = 0; i < b; i++)

cout<<inp[i][1]<<", ";

cout<<endl<<"Room - ";

for(int i = 0; i < c; i++)

cout<<inp[i][2]<<", ";

cout<<endl<<"Confirmed - ";

for(int i = 0; i < d; i++)

cout<<inp[i][3]<<",";

cout<<endl;

}

void selSort(vector<string> &inp){

for(int i = 0; i < inp.size() - 1; i++){

int indx = i;

string temp = inp[i];

for(int n = i+1; n<inp.size(); n++){

if(inp[n]<temp){

temp = inp[n];

indx = n;

}

}

inp[indx] = inp[i];

inp[i] = temp;

}

}

void bubSort(vector<string> &inp){

bool swap;

int n = inp.size();

do{

swap = false;

n--;

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

if(inp[i] > inp[i+1]){

swap = true;

string temp = inp[i];

inp[i] = inp[i+1];

inp[i+1] = temp;

}

}

}while(swap);

}

void setCha(string cha[]){

cha[0] = "Colonel Mustard";

cha[1] = "Miss Scarlet";

cha[2] = "Professor Plum";

cha[3] = "Mr. Green";

cha[4] = "Mrs. White";

cha[5] = "Mrs. Peacock";

}

void setWpon(string wpon[]){

wpon[0] = "Rope";

wpon[1] = "Lead Pipe";

wpon[2] = "Knife";

wpon[3] = "Wrench";

wpon[4] = "Candlestick";

wpon[5] = "Revolver";

}

void setRoom(string room[]){

room[0] = "Conservatory";

room[1] = "Billiard Room";

room[2] = "Library";

room[3] = "Study";

room[4] = "Ballroom";

room[5] = "Hall";

room[6] = "Kitchen";

room[7] = "Dining Room";

room[8] = "Lounge";

}

void suspect(vector<string> cha, vector<string> wpon, vector<string> room, vector<string> guilt, string spect[][4]){

if(cha.size()>1)

selSort(cha);

if(wpon.size()>1)

selSort(wpon);

if(room.size()>1)

bubSort(room);

if(guilt.size()>1)

bubSort(guilt);

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

spect[i][0] = cha[i];

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

spect[i][1] = wpon[i];

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

spect[i][2] = room[i];

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

spect[i][3] = guilt[i];

}

void readIn(fstream &in, string &spect){

string lEnd = "\*/";

//Display user suspect list

while(in>>spect){

if(spect == lEnd)

cout<<endl;

else

cout<<spect<<" ";

}

}

void readTo(fstream &out, string cha, string wpon, string room){

out<<"Suspect - "<<cha<<" /"<<endl;

out<<"Weapon - "<<wpon<<" /"<<endl;

out<<"Room - "<<room<<" /"<<endl;

}