Project 2

**Hearts (v3)**

**CIS-5 35411**

**Stephanie Peacock June 5, 2023**

**Introduction**

Title : Hearts

The game of Hearts is usually a four player game. Each player is dealt 13 cards (dividing the deck equally). The player’s intent is to score as low as possible. Each hand consists of thirteen tricks, in which each player plays one card. The lead player sets the suit for the trick, and remaining players must match the suit, if possible, else they can play the card of their choice.

The player with the two of clubs plays the first trick of each hand, usually point cards are not allowed during that trick, nor are hearts allowed to lead unless a heart has already been played. The winner of the trick (player with the highest card of the lead card suit) plays the first card of the next trick. This continues until all cards have been played.

Players score their hands as follows:

Any Heart card is worth one point. Queen of Spades is worth 13 points

If a player manages to collect all 14 point cards (for a total of 26 points) they score 0 points for the hand, and all remaining players are scored 26 points. This is called “Shooting the Moon.”

Players play until an agreed upon total point count is reached, usually 50 or 100. Once one player reaches that point, the game is over, and the player with the lowest point tally is considered the winner.

**Summary**

Project size: about 1200 lines

Major Variables:

* Structures: Player
  1. Members within:
  2. int order;
  3. string name;
  4. int hand[13];
  5. int choice;
  6. string cards[13];
  7. bool match;
  8. int tScore;
  9. int score;
* Arrays:
  1. String suit[]
  2. String show[]
  3. Char face[][]
* Vectors:
  1. <int> deck

My Modifications:

I really wanted to expand my game to all four players, and with the use of arrays and functions it made it possible to do. I followed most of the logic for the actual game. I did not enforce the player not using hearts in the opening trick, (which is possible if the player does not have any clubs), nor did I force the player to wait for hearts to be broken before beginning a trick with a heart (the Stooges would not do this unless they had no other suits in their hand).

I also kept the Stooges logic simpler than the actual game, with the exception of trying to ensure they can’t win the Queen of Spades when spades is played, and trying to ensure they will lose if hearts is played. If they do not have the suit in play, they will always choose to discard the Queen of Spades first, then the Ace of Spades, then the King of Spades, then the largest heart available. If they have the matching suit, Stooges will always play the lowest card in their hand, to avoid points.

Scoring is a matter of determining which player won the trick. This is done by checking each player to see if they won a trick with 4 hearts, 3 hearts, 2 hearts, 1 heart, or no hearts. For all ranks except 4 hearts, there is a variation that accounts for the Queen of spades being played, with points being allotted to the trick hand accordingly. Order is set with the winner being 0, and assigning all other players to 1 2 or 3, and all played choices are removed to prevent replays.

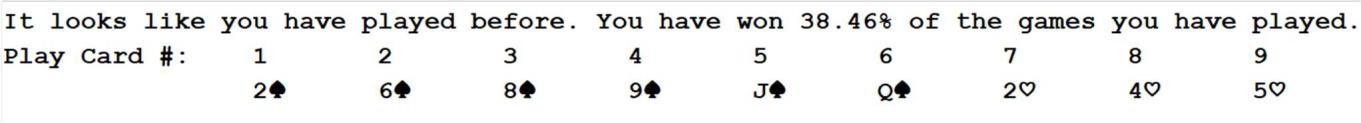
After all tricks has been scored, it is checked to see if any player Shot the Moon, if they did, points are assigned according by traditional rules. The trick score is added to the game score, and as long as no player lost (reached 50 points) the game loops back to play another hand. The player is given the option between hands to end the program early, but this does not affect the win/loss ratio, as they did not actually complete the game. If the player chooses to continue, the game loops until someone loses, and determines if the player won or lost. It displays their current win/loss ratio, saves the result to wins.txt and exits the program.

**Game Play Screenshots**

**This section is from Project 1, as gameplay does not vary, aside from displaying 4 played cards instead of 2. Some of the messages are more Stooge-like as well. I left those as surprises.**

A picture containing text, font, screenshot, white

Description automatically generatedThe game begins with an introduction that changes, depending upon if the player has ever played or not. If they have not, they are given a brief introduction to the game. If the player has played previously, they are told what percentage of games they have won so far.



A picture containing text, font, screenshot, white

Description automatically generatedFollowing the introduction, the player is shown their hand, and asked to choose a card. If they do not choose a card currently in their hand, or choose a card that has been previously played, they will be prompted to reenter their card.

As each trick is played, both player cards are shown, then scored. Played cards are removed from the player’s visible hand, so they don’t try to re-play them.

A picture containing text, screenshot, font, algebra

Description automatically generated

A picture containing text, receipt, font, screenshot

Description automatically generatedIf a player manages to shoot the moon, they are given a bonus message, and points are allocated to the computer instead.

A picture containing text, screenshot, font

Description automatically generated

A screenshot of a game

Description automatically generated with low confidenceAfter the game is finished the overall game score is shown, and the win % is displayed.

The player can’t always shoot the moon, however, so the usual scoring for the hand is displayed.

**Pseudocode**

// Program Open

// Add libraries, namespace and headers

// Start Main

// Set random seed

// Declare variables

// Open wins file

// Read in wins & total

// Close wins file

// Check if player has played before

// Make sure we don't divide by 0!

// If total !0

// Calculate win/loss ratio

// Output current win/loss ratio

// Else Introduce game, get player name

// Assign face values to the cards to match deck

// Check game score (GAME LOOP BEGINS) do while scores < 50

// initialize indx for linear search

// Set initial trick scores to 0 & match to false

// Shuffle deck & face values

// Deal Cards - 13 to each player, in order shuffled

// Sort player hands

// Bubble sort player

// Selection sort Larry

// Mark sort Curly & Moe

// Begin loop to play the hand (13 times - 1 per card in hand)

// reset index to 0

// If first trick find who has 2 clubs

//perform linear search to find 2 clubs

// While loop to go through the deck

// Increments the indx if not found

// ends the loop when 2 clubs selected

// return the indx 2 clubs was at

// assign initial player based on who has 2 clubs

// If indx is 0-12 player first, then larry, curly, moe

// If indx is 13-25 larry first, then curly, moe, player

// if indx is 26-38 curly first, then moe, player, larry

// else (indx 39-51) moe first, then player, larry, curly

// set match to false for all

// Begin loop to play trick (4 times - 1 per player)

// If player's turn (order = trick)

// Print cards

// Get player choice - do/while validate card is in range and has not been played

// If player has 2 clubs, force that play

// set player match to true

// Else if player not first, enforce suit compliance

// if player matched suit set to true, otherwise it stays false

// Else (player is first in the trick) player picks any card

// set match to true

// Print out what the player played

// Set the player's choice to = value of the hand[] they played for scoring

// If Stooge's turn (check to see if their count = trick)

// PlayCard program begins

// Set chosen to false (when choice is set chosen = true returns to main)

// Determine if Stooge order = 0

// If Stooge has 2 clubs, force that play

// Stooge first but no 2 clubs

// play smallest card

// If Player order = 0

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else If Larry order = 0 (& Stooge is not Larry)

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else If Curly order = 0 (& Stooge is not Curly)

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Else Moe order = 0

// Loop to Match the suit if possible - playing lowest card possible

// Loop to Play Q spades, A spades, or K spades first if can't match suit

// Loop to Lastly, play the highest card available

// Output the card Played

// Set the Stooge's choice to = value of the hand[] they played for scoring

// Score the trick

// all played hearts trick worth 4pts, largest card wins

//\*\*\*\*\*\*\*\*\*\*\*\*\* (process below is the same for 3 hearts, 2 hearts, no hearts)

// determine how many hearts were played

// find who matched the initial suit and had the highest card of that suit

// check if Q spades played

// 3 hearts: 16 pts, 2 hearts: 15 pts, 1 heart: 14 pts, no hearts: 13 pts

// output winner & assign trick score

// no Q spades played

// 3 hearts: 3pts, 2 hearts: 2 pts, 1 heart: 1 pt, 0 hearts: 0 pts

// output winner & assign trick score

// unset player choice from card value to 1-13

// increment hand with count();

// unset each Stooge choice from card value to 1-13

// end trick (trick increments)

// hand ends when hand has been played 13 times

// shuffle deck

// deal

// resort everyone's cards

// If anyone shoot moon - assign 26 points to all other opponents set that person's hand to 0

// Else add the hand score to the game score

// Output game score

// check if player wants to continue (do while to force choice to be y or no)

// if N then exit program

// check if player won & output result

// calculate player winLoss

// output winLoss %

// open wins, output wins & total, close the file

// END PROGRAM

**Flowcharts**

**A picture containing text, sketch, diagram, plan

Description automatically generated**

**A picture containing text, diagram, plan, sketch

Description automatically generated**

**Add more flowcharts here**

**Checklist**

**A close-up of a list

Description automatically generated with low confidence**

**A close-up of a document

Description automatically generated with low confidence**

**Reference**

1. Gaddis, T. (2019). *Starting out with C++ From Control Structures Through Objects* (9th ed.). Pearson Education Limited.
2. Savitch, W. J. (2018). *Problem solving with C++, Tenth edition / C++* (10th ed.). Pearson Education Limited. .
3. Class GitHub – Unique5CardPull\_V1\_ValuesInOrder, Unique5CardPull, FillFileWithCards

**Program Header File: Hearts.h**

#ifndef HEARTS\_H

#define HEARTS\_H

struct Player {

int order;

string name;

int hand[13];

int choice;

string cards[13];

bool match;

int tScore;

int score;

};

// Input function prototypes

void shuffle(vector<int>, const unsigned short, string[]);

void deal(Player &,Player &,Player &,Player &, vector<int>, string[]);

void bSort(Player &, int [], const short, string []);

void sSort(Player &, int [], const short, string []);

void mSort(Player &, int [], const short, string []);

void print(Player &, int [], string [], const short, int);

void deal(string [], char [][13], string [], const short, const short);

void set(Player &, int&, int[]);

void unset(Player &, int&, int[], string[]);

void npcUn(Player &, int &, int[], string[]);

void played(Player &, string, int, string[]);

int linSrch(vector<int>,const unsigned short,int&);

void playCard(Player&, Player&, Player&, Player&, int&, int&, int [], string, int, bool&);

void trick( Player&, Player&, Player&, Player&,

int, int, int, int,

int&, int&, int&, int&,

int&, int&, int&, int&,

bool ,bool ,bool ,bool);

void count();

#endif /\* HEARTS\_H \*/

**Hearts: main.cpp**

//System Libraries

#include <iostream> // Input / Output Library

#include <cstdlib> // Random Function Library

#include <ctime> // Time Library

#include <iomanip> // Formatting Library

#include <fstream> // File Library

#include <vector>

using namespace std;

//User Libraries

#include "hearts.h" // Hearts specific library

//Global Constants - Math/Physics/Chemistry/Conversions Only

//Function Prototypes - included in hearts.h

//Execution Begins Here

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

//Set Random number seed

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

//Declare Variable Data Types and Constants

const unsigned short NCARDS = 52; // 52 cards in a deck - no jokers

const short HAND = NCARDS/4; // divides the deck into 4

const short SUITS = 4;

// Array to hold the card number in the deck

vector<int> deck {

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, // indx 0 - 12

14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, // indx 13 - 25

27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, // indx 26 - 38

40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52};// indx 39 - 51

int indx;

int wins;

int total;

float winLoss;

char confirm;

// Displays Face Values of cards - Aces High

char face[SUITS][HAND] = {{'2','3','4','5','6','7','8','9','T','J','Q','K','A'},

{'2','3','4','5','6','7','8','9','T','J','Q','K','A'},

{'2','3','4','5','6','7','8','9','T','J','Q','K','A'},

{'2','3','4','5','6','7','8','9','T','J','Q','K','A'}};

// Displays the suit icon

string suit[SUITS] = {"\u2663", // Clubs

"\u2662", // Diamonds

"\u2660", // Spades

"\u2661"}; // Hearts

string show[NCARDS];

// Assigns a structure for each player

Player player, larry, curly, moe;

larry.name = "Larry";

curly.name = "Curly";

moe.name = "Moe";

player.score = larry.score = curly.score = moe.score = 0;

// Open file & read current win / total

ifstream inWins("wins.txt");

inWins >> wins >> total;

// Close the file for at the end!

inWins.close();

// Make sure we don't divide by 0!

if(total != 0 ) {

// Previous games have been played - output the ratio

winLoss = (wins/static\_cast<float>(total)) \* 100;

cout << "It looks like you have played before. You have won " << fixed << setprecision(2) << winLoss << "% of the games you have played." << endl;

}

else{

cout << "\"Hey buddy, we need a fourth player!" << endl

<< " Join us for a game of Hearts?" << endl

<< " My name is Larry, this is my good friend Curly," << endl

<< " and that's his brother, Moe.\"" << endl << endl;

cout << "\"So what's your name?\"" << endl;

cin >> player.name;

cout << "\"Alright " << player.name << ", let's play Hearts! I'll deal.\"" << endl;

}

// Assign face values

deal(show,face,suit,SUITS,HAND);

// put the outer do while loop for the game here

do {

//Initialize indx

indx = 0; // used for the linear search to find 2 of clubs - reset to 0 each hand

// Set initial trick scores to 0 & match to false

player.tScore = larry.tScore = curly.tScore = moe.tScore = 0;

// Shuffle the deck

shuffle(deck, NCARDS, show);

// Deal cards

deal(player, larry, curly, moe, deck, show);

//Sort each player's Hand

bSort(player, player.hand, HAND, player.cards);

sSort(larry, larry.hand, HAND, larry.cards);

mSort(curly, curly.hand, HAND, curly.cards);

mSort(moe, moe.hand, HAND, moe.cards);

// Loop to play the hand

for (int i = 0; i < 13; i++) {

// used for the linear search to find 2 of clubs - resets to 0 each hand

indx = 0;

// For the first trick, find the player with 2 of clubs

if(i == 0) {

linSrch(deck,NCARDS,indx);

if (indx < 13) { player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3; }

else if(indx > 12 && indx < 26) { player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2; }

else if(indx > 25 && indx < 39) { player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1; }

else { player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0; }

}

// reset match

player.match = larry.match = curly.match = moe.match = false;

// Loop 4 times with If/Else to play trick

for(int trick = 0; trick < 4; trick++){

//Player's Turn

if (player.order == trick) {

// Print player's cards

print(player, player.hand, player.cards, HAND, player.choice);

// Get player Choice - validates selection

do {

cout << "Choose a card in your hand you wish to play: ";

cin >> player.choice;

// Add 2clubs validation

while (player.order == 0 && player.hand[0] == 1 && player.choice != 1) {

cout << "Please play 2\u2663: ";

cin >> player.choice;

}

if(player.order == 0){

player.match = true;

}

// Add suit validation

// Check for Clubs

if((larry.order == 0 && larry.choice < 14) ||

(curly.order == 0 && curly.choice < 14) ||

(moe.order == 0 && moe.choice < 14)) {

for(int i = 0; i < 12; i++) {

int vc = i+1;

while(player.choice == vc && player.hand[i] != 0 && (player.hand[i] >13 && player.hand[i+1] < 13 )) {

cout << "Please play \u2663: ";

cin >> player.choice;

}

player.match = true;

}

}

// Check for Diamonds

if((larry.order == 0 && larry.choice > 13 && larry.choice < 27) ||

(curly.order == 0 && curly.choice > 13 && curly.choice < 27) ||

(moe.order == 0 && moe.choice > 13 && moe.choice < 27)) {

for(int i = 0; i < 12; i++) {

int vc = i+1;

while(player.choice == vc && player.hand[i] !=0 && (player.hand[i] <14 && player.hand[i] >26 && player.hand[i+1] >13 && player.hand[i+1] <27)) {

cout << "Please play \u2662: ";

cin >> player.choice;

}

player.match = true;

}

}

// Check for Spades

if((larry.order == 0 && larry.choice > 26 && larry.choice < 40) ||

(curly.order == 0 && curly.choice > 26 && curly.choice < 40) ||

(moe.order == 0 && moe.choice > 26 && moe.choice < 40)) {

for(int i = 0; i < 12; i++) {

int vc = i+1;

while(player.choice == vc && player.hand[i] !=0 && (player.hand[i] <27 && player.hand[i] >39 && player.hand[i+1] >26 && player.hand[i+1] <40)) {

cout << "Please play \u2660: ";

cin >> player.choice;

}

player.match = true;

}

}

// Check for hearts

if((larry.order == 0 && larry.choice > 39) ||

(curly.order == 0 && curly.choice > 39) ||

(moe.order == 0 && moe.choice > 39)) {

for(int i = 0; i < 12; i++) {

int vc = i+1;

while(player.choice == vc && player.hand[i] !=0 && (player.hand[i] <40 && player.hand[i+1] >39)) {

cout << "Please play \u2661: ";

cin >> player.choice;

}

player.match = true;

}

}

} while ((player.choice < 1 || player.choice > 13) ||

(player.choice == 1 && player.hand[0] == 0) ||

(player.choice == 2 && player.hand[1] == 0) ||

(player.choice == 3 && player.hand[2] == 0) ||

(player.choice == 4 && player.hand[3] == 0) ||

(player.choice == 5 && player.hand[4] == 0) ||

(player.choice == 6 && player.hand[5] == 0) ||

(player.choice == 7 && player.hand[6] == 0) ||

(player.choice == 8 && player.hand[7] == 0) ||

(player.choice == 9 && player.hand[8] == 0) ||

(player.choice == 10 && player.hand[9] == 0) ||

(player.choice == 11 && player.hand[10] == 0) ||

(player.choice == 12 && player.hand[11] == 0) ||

(player.choice == 13 && player.hand[12] == 0));

// Valid card chosen, set match to true

player.match = true;

// Output the card played

played(player, player.name, player.choice, player.cards);

// Set choice value to the played card value for scoring

set(player, player.choice, player.hand);

// Larry's Turn

} else if (larry.order == trick) {

// cout << "Larry's cards";

// print(larry, larry.hand, larry.cards, HAND, larry.choice, count);

playCard(larry,curly,moe, player, larry.choice, larry.order, larry.hand, larry.name, trick, larry.match);

// Output the card played

played(larry, larry.name, larry.choice, larry.cards);

// Set the played card for scoring

set(larry, larry.choice, larry.hand);

// Curly's Turn

}else if(curly.order == trick) {

// cout << "Curly's cards";

// print(curly, curly.hand, curly.cards, HAND, curly.choice, count);

playCard(larry,curly,moe, player, curly.choice, curly.order, curly.hand, curly.name, trick, curly.match);

// Output the card played

played(curly, curly.name, curly.choice, curly.cards);

// Set the played card for scoring

set(curly, curly.choice, curly.hand);

// Moe's Turn

}else if(moe.order == trick) {

// cout << "Moe's cards"<< endl;

// print(moe, moe.hand, moe.cards, HAND, moe.choice, count);

playCard(larry,curly,moe, player, moe.choice, moe.order, moe.hand, moe.name, trick, moe.match);

// Output the card played

played(moe, moe.name, moe.choice, moe.cards);

// Set the played card for scoring

set(moe, moe.choice, moe.hand);

}

}

// Score Trick & set the player order for the next trick

trick(player,larry,curly,moe,

player.choice,larry.choice,curly.choice,moe.choice,

player.tScore,larry.tScore,curly.tScore,moe.tScore,

player.order,larry.order,curly.order,moe.order,

player.match,larry.match,curly.match,moe.match);

// Change choice back it's original value and remove played cards

unset(player,player.choice, player.hand, player.cards);

count();

npcUn(larry,larry.choice, larry.hand, larry.cards);

npcUn(curly,curly.choice, curly.hand, curly.cards);

npcUn(moe,moe.choice, moe.hand, moe.cards);

}

// Shuffle the deck

shuffle(deck, NCARDS, show);

// Deal cards

deal(player, larry, curly, moe, deck, show);

//Sort each player's Hand

bSort(player, player.hand, HAND, player.cards);

sSort(larry, larry.hand, HAND, larry.cards);

mSort(curly, curly.hand, HAND, curly.cards);

mSort(moe, moe.hand, HAND, moe.cards);

// Check for Shooting the Moon

player.tScore == 26 ? player.tScore = 0, larry.tScore = 26, curly.tScore = 26, moe.tScore = 26,cout << "Hey, Wiseguy! You shot the moon! The Stooges takes all points." << endl << endl :

larry.tScore == 26 ? larry.tScore = 0, player.tScore = 26, curly.tScore = 26, moe.tScore = 26, cout << "Larry shot the moon! The rest of you schmucks can take the points." << endl << endl :

curly.tScore == 26 ? curly.tScore = 0, player.tScore = 26, larry.tScore = 26, moe.tScore = 26, cout << "Curly shot the moon! The rest of you schmucks can take the points." << endl << endl :

moe.tScore == 26 ? moe.tScore = 0, player.tScore = 26, curly.tScore = 26, larry.tScore = 26, cout << "Larry shot the moon! The rest of you schmucks can take the points." << endl << endl :

cout << endl << "What a fun hand!" << endl << endl;

player.score += player.tScore;

larry.score += larry.tScore;

curly.score += curly.tScore;

moe.score += moe.tScore;

cout << "The current game score is:" << endl

<< "Player : " << player.score << endl

<< "Larry : " << larry.score << endl

<< "Curly : " << curly.score << endl

<< "Moe : " << moe.score << endl ;

cout << "Do you want to continue the game? Y or N";

do{

cin >> confirm;

if(confirm == 'N' or confirm == 'n') {

exit(0);

}

}while(confirm != 'Y' && confirm != 'y' && confirm != 'N' && confirm != 'n');

} while (player.score < 50 && larry.score < 50 && curly.score < 50 && moe.score <50);

cout << endl;

if (player.score < larry.score && player.score < curly.score && player.score < moe.score) { wins++, cout << "You won the game!" << endl; }

else {total++, cout << "You lost the game. Better luck next time!" << endl; }

// Let them know how good they are

winLoss = (wins/static\_cast<float>(total)) \* 100;

cout << "You have won " << fixed << setprecision(2) << winLoss << "% of the games you have played." << endl;

// Record win ratio to file

ofstream outWins("wins.txt");

outWins << wins << " " << total;

outWins.close();

return 0;

}

**Begin Functions**

void deal(string show[], char face[][13], string suit[], const short SUITS, const short HAND) {

int cFace = 0;

for(int sym = 0; sym < SUITS; sym++) {

for(int rank = 0; rank < HAND; rank++) {

show[cFace] = face[sym][rank];

show[cFace] += suit[sym] + "\t";

++cFace;

}

}

}

void shuffle(vector<int> deck, const unsigned short NCARDS, string show[]) {

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

int n = i + (rand()%(52 - i));

swap(deck[i], deck[n]);

swap(show[i], show[n]);

}

}

void deal(Player &player,Player &larry,Player &curly,Player &moe, vector<int>deck, string show[]){

for(int i = 0; i < 13; i++) {

player.hand[i] = deck[i];

larry.hand[i] = deck[(i+13)];

curly.hand[i] = deck[(i+26)];

moe.hand[i] = deck[(i+39)];

player.cards[i] = show[i];

larry.cards[i] = show[(i+13)];

curly.cards[i] = show[(i+26)];

moe.cards[i] = show[(i+39)];

}

}

void bSort(Player &, int hand[], const short HAND, string cards[]) {

bool swap;

do {

swap = false;

int n = 1;

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

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

int temp = hand[i];

hand[i] = hand[i+1];

hand[i+1] = temp;

string tmp = cards[i];

cards[i] = cards[i+1];

cards[i+1] = tmp;

swap = true;

}

}

n--;

} while (swap);

}

void sSort(Player &, int hand[], const short HAND, string cards[]) {

int indx;

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

indx = i;

for(int j = i+1; j < HAND; j++)

if (hand[j] < hand[indx])

indx = j;

swap(hand[i],hand[indx]);

swap(cards[i],cards[indx]);

}

}

void mSort(Player &, int hand[], const short HAND, string cards[]){

for(int j = 0; j < HAND - 1; j++){

for(int i = j + 1; i < HAND; i++){

if(hand[j] > hand[i]){

int temp = hand[j];

hand[j] = hand[i];

hand[i] = temp;

string tmp = cards[j];

cards[j] = cards[i];

cards[i] = tmp;

}

}

}

}

int linSrch(vector<int> deck,const unsigned short NCARDS,int& indx) {

// start off with false

bool found = false;

// run until the val is found or we run through all the numbers

while (indx < NCARDS && !found){

// if the val is found it returns true and stops the loop, if false it adds to the count

deck[indx] == 1 ? found = true : indx++;

}

// sends back what index 2 of clubs (1) was found at

return indx;

}

void print(Player &, int hand[], string cards[], const short HAND, int choice) {

if (count) {

// output Player's remaining cards - if value is 0 an empty string is output

cout << endl << "Play Card #:\t";

if (hand[0] == 0) { cout << ""; } else { cout << "1" << "\t"; }

if (hand[1] == 0) { cout << ""; } else { cout << "2" << "\t"; }

if (hand[2] == 0) { cout << ""; } else { cout << "3" << "\t"; }

if (hand[3] == 0) { cout << ""; } else { cout << "4" << "\t"; }

if (hand[4] == 0) { cout << ""; } else { cout << "5" << "\t"; }

if (hand[5] == 0) { cout << ""; } else { cout << "6" << "\t"; }

if (hand[6] == 0) { cout << ""; } else { cout << "7" << "\t"; }

if (hand[7] == 0) { cout << ""; } else { cout << "8" << "\t"; }

if (hand[8] == 0) { cout << ""; } else { cout << "9" << "\t"; }

if (hand[9] == 0) { cout << ""; } else { cout << "10"<< "\t"; }

if (hand[10]== 0) { cout << ""; } else { cout << "11"<< "\t"; }

if (hand[11]== 0) { cout << ""; } else { cout << "12"<< "\t"; }

if (hand[12]== 0) { cout << ""; } else { cout << "13"<< "\t"; }

// now output the remaining face values

cout << endl << setw(14) << "\t";

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

cout << cards[j];

}

cout << endl;

}

}

void played(Player &, string name, int choice, string cards[]) {

cout << name << " played: ";

switch (choice) {

case 1: cout << cards[0]; break;

case 2: cout << cards[1]; break;

case 3: cout << cards[2]; break;

case 4: cout << cards[3]; break;

case 5: cout << cards[4]; break;

case 6: cout << cards[5]; break;

case 7: cout << cards[6]; break;

case 8: cout << cards[7]; break;

case 9: cout << cards[8]; break;

case 10:cout << cards[9]; break;

case 11:cout << cards[10];break;

case 12:cout << cards[11];break;

default:cout << cards[12];break;

}

cout << endl;

}

void set(Player &, int &choice, int hand[] ){

// Set choice to card value for scoring

choice == 1 ? choice = hand[0] : choice == 2 ? choice = hand[1] :

choice == 3 ? choice = hand[2] : choice == 4 ? choice = hand[3] :

choice == 5 ? choice = hand[4] : choice == 6 ? choice = hand[5] :

choice == 7 ? choice = hand[6] : choice == 8 ? choice = hand[7] :

choice == 9 ? choice = hand[8] : choice == 10 ? choice = hand[9] :

choice == 11 ? choice = hand[10]: choice == 12 ? choice = hand[11]:

choice = hand[12];

}

void playCard(Player& larry,Player& curly,Player& moe, Player& player, int& choice, int & order, int hand[], string name, int trick, bool& match) {

bool chosen = false;

// If npc is first to play in the trick - check for 2 clubs first

if(order == 0) {

if ( hand[0] == 1) {choice = 1; match = true;}

// If no 2 clubs, play lowest card

else if (order == 0){

for(int i = 0; i < 13 && !chosen; i++){

if(hand[i] != 0) { choice = i+1; chosen = true; match = true; }

}

}

}

else {

// Check against player first

if(player.order == 0) {

int min = player.choice < 14 ? 0 : player.choice < 27 ? 14 : player.choice < 40 ? 27 : 40;

int max = min + 13;

// Match the suit if possible - playing lowest card possible

for (int i = 0; i < 13 && !chosen; i++) {

if (hand[i] >= min && hand[i] <= max && hand[i] != 0) {

choice = (i+1); chosen = true; match = true; }

}

// Play Q spades, A spades, or K spades first if can't match suit

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] == 37) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 39) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 38) {choice = (j+1); chosen = true; match = false;}

}

}

// Lastly, play the highest card available

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] != 0) {choice = (j+1); chosen = true; match = false; }

}

}

}

// Check against Larry second

else if(larry.order == 0 && order != larry.order) {

int min = larry.choice < 14 ? 0 : larry.choice < 27 ? 14 : larry.choice < 40 ? 27 : 40;

int max = min + 13;

// Match the suit if possible - playing lowest card possible

for (int i = 0; i < 13 && !chosen; i++) {

if (hand[i] >= min && hand[i] <= max && hand[i] != 0) {

choice = (i+1); chosen = true; match = true; }

}

// Play Q spades, A spades, or K spades first if can't match suit

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] == 37) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 39) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 38) {choice = (j+1); chosen = true; match = false;}

}

}

// Lastly, play the highest card available

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] != 0) {choice = (j+1); chosen = true; match = false; }

}

}

}

// Check against Curly third

else if(curly.order == 0 && order != curly.order) {

int min = curly.choice < 14 ? 0 : curly.choice < 27 ? 14 : curly.choice < 40 ? 27 : 40;

int max = min + 13;

// Match the suit if possible - playing lowest card possible

for (int i = 0; i < 13 && !chosen; i++) {

if (hand[i] >= min && hand[i] <= max && hand[i] != 0) {

choice = (i+1); chosen = true; match = true; }

}

// Play Q spades, A spades, or K spades first if can't match suit

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] == 37) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 39) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 38) {choice = (j+1); chosen = true; match = false;}

}

}

// Lastly, play the highest card available

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] != 0) {choice = (j+1); chosen = true; match = false; }

}

}

}

// Check against Moe Last

else {

int min = moe.choice < 14 ? 0 : moe.choice < 27 ? 14 : moe.choice < 40 ? 27 : 40;

int max = min + 13;

// Match the suit if possible - playing lowest card possible

for (int i = 0; i < 13 && !chosen; i++) {

if (hand[i] >= min && hand[i] <= max && hand[i] != 0) {

choice = (i+1); chosen = true; match = true; }

}

// Play Q spades, A spades, or K spades first if can't match suit

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] == 37) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 39) {choice = (j+1); chosen = true; match = false;}

if (hand[j] == 38) {choice = (j+1); chosen = true; match = false;}

}

}

// Lastly, play the highest card available

if(!chosen){

for (int j = 12; j > 0 && !chosen; j--) {

if (hand[j] != 0) {choice = (j+1); chosen = true; match = false; }

} }} }}

void unset(Player &, int &choice, int hand[], string cards[]){

// Set choice back to the card number instead of card value & set used card to 0

choice == hand[0] ? choice = 1 : choice == hand[1] ? choice = 2 :

choice == hand[2] ? choice = 3 : choice == hand[3] ? choice = 4 :

choice == hand[4] ? choice = 5 : choice == hand[5] ? choice = 6 :

choice == hand[6] ? choice = 7 : choice == hand[7] ? choice = 8 :

choice == hand[8] ? choice = 9 : choice == hand[9] ? choice = 10 :

choice == hand[10] ? choice = 11 : choice == hand[11] ? choice = 12 :

choice = 13;

// set chosen card to 0

if (choice == 1) { hand[0] = 0; cards[0] = "";}

if (choice == 2) { hand[1] = 0; cards[1] = "";}

if (choice == 3) { hand[2] = 0; cards[2] = "";}

if (choice == 4) { hand[3] = 0; cards[3] = "";}

if (choice == 5) { hand[4] = 0; cards[4] = "";}

if (choice == 6) { hand[5] = 0; cards[5] = "";}

if (choice == 7) { hand[6] = 0; cards[6] = "";}

if (choice == 8) { hand[7] = 0; cards[7] = "";}

if (choice == 9) { hand[8] = 0; cards[8] = "";}

if (choice == 10){ hand[9] = 0; cards[9] = "";}

if (choice == 11){ hand[10] = 0; cards[10] = "";}

if (choice == 12){ hand[11] = 0; cards[11] = "";}

if (choice == 13){ hand[12] = 0; cards[12] = "";}

}

void npcUn(Player &, int &choice, int hand[], string cards[]){

// Set choice back to the card number instead of card value & set used card to 0

choice == hand[0] ? choice = 1 : choice == hand[1] ? choice = 2 :

choice == hand[2] ? choice = 3 : choice == hand[3] ? choice = 4 :

choice == hand[4] ? choice = 5 : choice == hand[5] ? choice = 6 :

choice == hand[6] ? choice = 7 : choice == hand[7] ? choice = 8 :

choice == hand[8] ? choice = 9 : choice == hand[9] ? choice = 10 :

choice == hand[10] ? choice = 11 : choice == hand[11] ? choice = 12 :

choice = 13;

// set chosen card to 0

if (choice == 1) { hand[0] = 0; cards[0] = ""; }

if (choice == 2) { hand[1] = 0; cards[1] = ""; }

if (choice == 3) { hand[2] = 0; cards[2] = ""; }

if (choice == 4) { hand[3] = 0; cards[3] = ""; }

if (choice == 5) { hand[4] = 0; cards[4] = ""; }

if (choice == 6) { hand[5] = 0; cards[5] = ""; }

if (choice == 7) { hand[6] = 0; cards[6] = ""; }

if (choice == 8) { hand[7] = 0; cards[7] = ""; }

if (choice == 9) { hand[8] = 0; cards[8] = ""; }

if (choice == 10){ hand[9] = 0; cards[9] = ""; }

if (choice == 11){ hand[10] = 0; cards[10] = ""; }

if (choice == 12){ hand[11] = 0; cards[11] = ""; }

if (choice == 13){ hand[12] = 0; cards[12] = ""; }

}

void trick( Player& player, Player& larry, Player& curly, Player& moe,

int playerchoice,int larrychoice,int curlychoice,int moechoice,

int& playertScore, int& larrytScore,int& curlytScore,int& moetScore,

int& playerorder,int& larryorder,int& curlyorder,int& moeorder,

bool playermatch,bool larrymatch,bool curlymatch,bool moematch){

// All Play Hearts

if (player.choice > 39 && larry.choice > 39 && curly.choice > 39 && moe.choice > 39) {

cout << "Trick worth four points." << endl;

// Player wins trick

if (player.choice > larry.choice && player.choice > curly.choice && player.choice > moe.choice) {

cout << "Player takes the trick." << endl;

player.tScore += 4;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

// Larry wins trick

else if (larry.choice > player.choice && larry.choice > curly.choice && larry.choice > moe.choice){

cout << "Larry takes the trick." << endl;

larry.tScore += 4;

player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2;

}

// Curly wins trick

else if (curly.choice > player.choice && curly.choice > larry.choice && curly.choice > moe.choice){

cout << "Curly takes the trick." << endl;

curly.tScore += 4;

player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1;

}

// Moe wins trick

else {

cout << "Moe takes the trick." << endl;

larry.tScore += 4;

player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0;

}

}

// Three Play Hearts

else if ((player.choice > 39 && larry.choice > 39 && curly.choice > 39) ||

(player.choice > 39 && larry.choice > 39 && moe.choice > 39) ||

(player.choice > 39 && curly.choice > 39 && moe.choice > 39) ||

(larry.choice > 39 && curly.choice > 39 && moe.choice > 39)){

// No Queen Spades Played

if(player.choice != 37 && larry.choice != 37 && curly.choice != 37 && moe.choice != 37) {

cout << "Trick worth three points." << endl;

}

// Queen of Spades Played

else if(larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ) {

cout << "Trick worth sixteen points." << endl;

}

// Player wins trick

if((player.order == 0 && larry.match == true && curly.match == true && player.choice > larry.choice && player.choice > curly.choice) ||

(player.order == 0 && curly.match == true && moe.match == true && player.choice > curly.choice && player.choice > moe.choice) ||

(player.order == 0 && larry.match == true && moe.match == true && player.choice > larry.choice && player.choice > moe.choice) ||

(larry.order = 0 && curly.match == true && player.match == true && player.choice > larry.choice && player.choice > curly.choice) ||

(larry.order = 0 && moe.match == true && player.match == true &&player.choice > larry.choice && player.choice > moe.choice) ||

(curly.order = 0 && larry.match == true && player.match == true &&player.choice > larry.choice && player.choice > curly.choice) ||

(curly.order = 0 && moe.match == true && player.match == true &&player.choice > curly.choice && player.choice > curly.choice) ||

(moe.order = 0 && curly.match == true && player.match == true &&player.choice > moe.choice && player.choice > curly.choice) ||

(moe.order = 0 && larry.match == true && player.match == true &&player.choice > larry.choice && player.choice > moe.choice)) {

cout << "Player takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37){

player.tScore += 16;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

// No one played Q spades

else {

player.tScore += 3;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

}

// Larry wins trick

else if((larry.order == 0 && player.match == true && curly.match == true && larry.choice > player.choice && larry.choice > curly.choice) ||

(larry.order == 0 && curly.match == true && moe.match == true && larry.choice > curly.choice && larry.choice > moe.choice) ||

(larry.order == 0 && player.match == true && moe.match == true && larry.choice > player.choice && larry.choice > moe.choice) ||

(player.order = 0 && curly.match == true && larry.match == true && larry.choice > player.choice && larry.choice > curly.choice) ||

(player.order = 0 && moe.match == true && larry.match == true && larry.choice > player.choice && larry.choice > moe.choice) ||

(curly.order = 0 && player.match == true && larry.match == true && larry.choice > player.choice && larry.choice > curly.choice) ||

(curly.order = 0 && moe.match == true && larry.match == true && larry.choice > curly.choice && larry.choice > curly.choice) ||

(moe.order = 0 && curly.match == true && larry.match == true && larry.choice > moe.choice && larry.choice > curly.choice) ||

(moe.order = 0 && player.match == true && larry.match == true && larry.choice > player.choice && larry.choice > moe.choice)) {

cout << "larry takes the trick." << endl;

// Check for Q spades

if (player.choice == 37 || curly.choice == 37 || moe.choice == 37 || larry.choice == 37){

larry.tScore += 16;

larry.order = 0; player.order = 1; curly.order = 2; moe.order = 3;

}

// No one played Q spades

else {

larry.tScore += 3;

larry.order = 0; player.order = 1; curly.order = 2; moe.order = 3;

}

}

// curly wins trick

else if((curly.order == 0 && player.match == true && larry.match == true && curly.choice > player.choice && curly.choice > larry.choice) ||

(curly.order == 0 && larry.match == true && moe.match == true && curly.choice > larry.choice && curly.choice > moe.choice) ||

(curly.order == 0 && player.match == true && moe.match == true && curly.choice > player.choice && curly.choice > moe.choice) ||

(player.order = 0 && larry.match == true && curly.match == true && curly.choice > player.choice && curly.choice > larry.choice) ||

(player.order = 0 && moe.match == true && curly.match == true && curly.choice > player.choice && curly.choice > moe.choice) ||

(larry.order = 0 && player.match == true && curly.match == true && curly.choice > player.choice && curly.choice > larry.choice) ||

(larry.order = 0 && moe.match == true && curly.match == true && curly.choice > larry.choice && curly.choice > larry.choice) ||

(moe.order = 0 && larry.match == true && curly.match == true && curly.choice > moe.choice && curly.choice > larry.choice) ||

(moe.order = 0 && player.match == true && curly.match == true && curly.choice > player.choice && curly.choice > moe.choice)) {

cout << "curly takes the trick." << endl;

// Check for Q spades

if (player.choice == 37 || larry.choice == 37 || moe.choice == 37 || curly.choice == 37){

curly.tScore += 16;

curly.order = 0; player.order = 1; larry.order = 2; moe.order = 3;

}

// No one played Q spades

else {

curly.tScore += 3;

curly.order = 0; player.order = 1; larry.order = 2; moe.order = 3;

}

}

// moe wins trick

else if((moe.order == 0 && player.match == true && larry.match == true && moe.choice > player.choice && moe.choice > larry.choice) ||

(moe.order == 0 && larry.match == true && curly.match == true && moe.choice > larry.choice && moe.choice > curly.choice) ||

(moe.order == 0 && player.match == true && curly.match == true && moe.choice > player.choice && moe.choice > curly.choice) ||

(player.order = 0 && larry.match == true && moe.match == true && moe.choice > player.choice && moe.choice > larry.choice) ||

(player.order = 0 && curly.match == true && moe.match == true && moe.choice > player.choice && moe.choice > curly.choice) ||

(larry.order = 0 && player.match == true && moe.match == true && moe.choice > player.choice && moe.choice > larry.choice) ||

(larry.order = 0 && curly.match == true && moe.match == true && moe.choice > larry.choice && moe.choice > larry.choice) ||

(curly.order = 0 && larry.match == true && moe.match == true && moe.choice > curly.choice && moe.choice > larry.choice) ||

(curly.order = 0 && player.match == true && moe.match == true && moe.choice > player.choice && moe.choice > curly.choice)) {

cout << "moe takes the trick." << endl;

// Check for Q spades

if (player.choice == 37 || larry.choice == 37 || curly.choice == 37 || moe.choice == 37){

moe.tScore += 16;

moe.order = 0; player.order = 1; larry.order = 2; curly.order = 3;

}

// No one played Q spades

else {

moe.tScore += 3;

moe.order = 0; player.order = 1; larry.order = 2; curly.order = 3;

}

}

}

// Two Play Hearts

else if ((player.choice > 39 && larry.choice > 39) ||

(player.choice > 39 && curly.choice > 39) ||

(player.choice > 39 && moe.choice > 39) ||

(larry.choice > 39 && curly.choice > 39) ||

(larry.choice > 39 && moe.choice > 39) ||

(curly.choice > 39 && moe.choice > 39)){

// No Queen Spades played

if(player.choice != 37 && larry.choice != 37 && curly.choice != 37 && moe.choice != 37) {

cout << "Trick worth two points." << endl;

}

else if(larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ) {

cout << "Trick worth fifteen points." << endl;

}

// player wins trick

if((player.order == 0 && player.choice > 39 && larry.choice > 39 && player.choice > larry.choice) ||

(player.order == 0 && player.choice > 39 && curly.choice > 39 && player.choice > curly.choice) ||

(player.order == 0 && player.choice > 39 && moe.choice > 39 && player.choice > moe.choice) ||

(player.order == 0 && player.choice < 40 && larry.choice > 39 && curly.choice > 39) ||

(player.order == 0 && player.choice < 40 && larry.choice > 39 && moe.choice > 39) ||

(player.order == 0 && player.choice < 40 && moe.choice > 39 && curly.choice > 39) ||

(moe.order == 0 && moe.choice > 39 && player.match == true && player.choice > moe.choice) ||

(larry.order == 0 && larry.choice > 39 && player.match == true && player.choice > larry.choice) ||

(curly.order == 0 && curly.choice > 39 && player.match == true && player.choice > curly.choice)) {

cout << "player takes the trick." << endl;

// Check for Q spades

if (moe.choice == 37 || larry.choice == 37 || curly.choice == 37 || player.choice == 37){

player.tScore += 15;

player.order = 0; moe.order = 1; larry.order = 2; curly.order = 3;

}

// No one played Q spades

else {

player.tScore += 2;

player.order = 0; moe.order = 1; larry.order = 2; curly.order = 3;

}

}

// larry wins trick

else if((larry.order == 0 && larry.choice > 39 && player.choice > 39 && larry.choice > player.choice) ||

(larry.order == 0 && larry.choice > 39 && curly.choice > 39 && larry.choice > curly.choice) ||

(larry.order == 0 && larry.choice > 39 && moe.choice > 39 && larry.choice > moe.choice) ||

(larry.order == 0 && larry.choice < 40 && player.choice > 39 && curly.choice > 39) ||

(larry.order == 0 && larry.choice < 40 && player.choice > 39 && moe.choice > 39) ||

(larry.order == 0 && larry.choice < 40 && moe.choice > 39 && curly.choice > 39) ||

(moe.order == 0 && moe.choice > 39 && larry.match == true && larry.choice > moe.choice) ||

(player.order == 0 && player.choice > 39 && larry.match == true && larry.choice > player.choice) ||

(curly.order == 0 && curly.choice > 39 && larry.match == true && larry.choice > curly.choice)) {

cout << "larry takes the trick." << endl;

// Check for Q spades

if (moe.choice == 37 || player.choice == 37 || curly.choice == 37 || larry.choice == 37){

larry.tScore += 15;

larry.order = 0; moe.order = 1; player.order = 2; curly.order = 3;

}

// No one played Q spades

else {

larry.tScore += 2;

larry.order = 0; moe.order = 1; player.order = 2; curly.order = 3;

}

}

// curly wins trick

else if((curly.order == 0 && curly.choice > 39 && player.choice > 39 && curly.choice > player.choice) ||

(curly.order == 0 && curly.choice > 39 && larry.choice > 39 && curly.choice > larry.choice) ||

(curly.order == 0 && curly.choice > 39 && moe.choice > 39 && curly.choice > moe.choice) ||

(curly.order == 0 && curly.choice < 40 && player.choice > 39 && larry.choice > 39) ||

(curly.order == 0 && curly.choice < 40 && player.choice > 39 && moe.choice > 39) ||

(curly.order == 0 && curly.choice < 40 && moe.choice > 39 && larry.choice > 39) ||

(moe.order == 0 && moe.choice > 39 && curly.match == true && curly.choice > moe.choice) ||

(player.order == 0 && player.choice > 39 && curly.match == true && curly.choice > player.choice) ||

(larry.order == 0 && larry.choice > 39 && curly.match == true && curly.choice > larry.choice)) {

cout << "curly takes the trick." << endl;

// Check for Q spades

if (moe.choice == 37 || player.choice == 37 || larry.choice == 37 || curly.choice == 37){

curly.tScore += 15;

curly.order = 0; moe.order = 1; player.order = 2; larry.order = 3;

}

// No one played Q spades

else {

curly.tScore += 2;

curly.order = 0; moe.order = 1; player.order = 2; larry.order = 3;

}

}

// moe wins trick

else if((moe.order == 0 && moe.choice > 39 && player.choice > 39 && moe.choice > player.choice) ||

(moe.order == 0 && moe.choice > 39 && larry.choice > 39 && moe.choice > larry.choice) ||

(moe.order == 0 && moe.choice > 39 && curly.choice > 39 && moe.choice > curly.choice) ||

(moe.order == 0 && moe.choice < 40 && player.choice > 39 && larry.choice > 39) ||

(moe.order == 0 && moe.choice < 40 && player.choice > 39 && curly.choice > 39) ||

(moe.order == 0 && moe.choice < 40 && curly.choice > 39 && larry.choice > 39) ||

(curly.order == 0 && curly.choice > 39 && moe.match == true && moe.choice > curly.choice) ||

(player.order == 0 && player.choice > 39 && moe.match == true && moe.choice > player.choice) ||

(larry.order == 0 && larry.choice > 39 && moe.match == true && moe.choice > larry.choice)) {

cout << "moe takes the trick." << endl;

// Check for Q spades

if (curly.choice == 37 || player.choice == 37 || larry.choice == 37 || moe.choice == 37){

moe.tScore += 15;

moe.order = 0; curly.order = 1; player.order = 2; larry.order = 3;

}

// No one played Q spades

else {

moe.tScore += 2;

moe.order = 0; curly.order = 1; player.order = 2; larry.order = 3;

}

}

}

// One person played a heart

else if (player.choice > 39 || larry.choice > 39 || curly.choice > 39 || moe.choice > 39) {

// No Queen spades played

if(player.choice != 37 && larry.choice != 37 && curly.choice != 37 && moe.choice != 37) {

cout << "Trick worth one point." << endl;

}

else if(larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ) {

cout << "Trick worth fourteen points." << endl;

}

// Player wins trick

if((player.order == 0 && player.choice > 39 && larry.choice < 40 && curly.choice < 40 && moe.choice < 40) ||

(player.order == 0 && player.choice < 40 && larry.choice > 39 && (curly.match == true && moe.match == true && player.choice > curly.choice && player.choice > moe.choice) || (curly.match == true && moe.match == false && moe.choice < 40 && player.choice > curly.choice) || (moe.match == true && curly.match == false && curly.choice < 40 && player.choice > moe.choice)) ||

(player.order == 0 && player.choice < 40 && curly.choice > 39 && (larry.match == true && moe.match == true && player.choice > larry.choice && player.choice > moe.choice) || (larry.match == true && moe.match == false && moe.choice < 40 && player.choice > larry.choice) || (moe.match == true && larry.match == false && larry.choice < 40 && player.choice > moe.choice)) ||

(player.order == 0 && player.choice < 40 && moe.choice > 39 && (larry.match == true && curly.match == true && player.choice > larry.choice && player.choice > curly.choice) || (larry.match == true && curly.match == false && curly.choice < 40 && player.choice > larry.choice) || (curly.match == true && larry.match == false && larry.choice < 40 && player.choice > curly.choice)) ||

(larry.order == 0 && larry.choice < 40 && player.match == true && moe.match == true && curly.choice > 39 && player.choice > larry.choice && player.choice > moe.choice) ||

(larry.order == 0 && larry.choice < 40 && player.match == true && moe.match == false && moe.choice < 40 && curly.choice > 39 && player.choice > larry.choice && player.choice > moe.choice) ||

(larry.order == 0 && larry.choice < 40 && player.match == true && curly.match == true && moe.choice > 39 && player.choice > larry.choice && player.choice > curly.choice)||

(larry.order == 0 && larry.choice < 40 && player.match == true && curly.match == false && curly.choice < 40 && moe.choice > 39 && player.choice > larry.choice && player.choice > curly.choice)||

(curly.order == 0 && curly.choice < 40 && player.match == true && moe.match == true && larry.choice > 39 && player.choice > curly.choice && player.choice > moe.choice) ||

(curly.order == 0 && curly.choice < 40 && player.match == true && moe.match == false && moe.choice < 40 && larry.choice > 39 && player.choice > curly.choice && player.choice > moe.choice) ||

(curly.order == 0 && curly.choice < 40 && player.match == true && larry.match == true && moe.choice > 39 && player.choice > curly.choice && player.choice > larry.choice)||

(curly.order == 0 && curly.choice < 40 && player.match == true && larry.match == false && larry.choice < 40 && moe.choice > 39 && player.choice > curly.choice && player.choice > larry.choice)||

(moe.order == 0 && moe.choice < 40 && player.match == true && curly.match == true && larry.choice > 39 && player.choice > moe.choice && player.choice > curly.choice) ||

(moe.order == 0 && moe.choice < 40 && player.match == true && curly.match == false && curly.choice < 40 && larry.choice > 39 && player.choice > moe.choice && player.choice > curly.choice) ||

(moe.order == 0 && moe.choice < 40 && player.match == true && larry.match == true && curly.choice > 39 && player.choice > moe.choice && player.choice > larry.choice)||

(moe.order == 0 && moe.choice < 40 && player.match == true && larry.match == false && larry.choice < 40 && curly.choice > 39 && player.choice > moe.choice && player.choice > larry.choice))

{

cout << "Player takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37){

player.tScore += 14;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

// No one played Q spades

else {

player.tScore += 1;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

}

// Larry wins trick

else if((larry.order == 0 && larry.choice > 39 && player.choice < 40 && curly.choice < 40 && moe.choice < 40) ||

(larry.order == 0 && larry.choice < 40 && player.choice > 39 && (curly.match == true && moe.match == true && larry.choice > curly.choice && larry.choice > moe.choice) || (curly.match == true && moe.match == false && moe.choice < 40 && larry.choice > curly.choice) || (moe.match == true && curly.match == false && curly.choice < 40 && larry.choice > moe.choice)) ||

(larry.order == 0 && larry.choice < 40 && curly.choice > 39 && (player.match == true && moe.match == true && larry.choice > player.choice && larry.choice > moe.choice) || (player.match == true && moe.match == false && moe.choice < 40 && larry.choice > player.choice) || (moe.match == true && player.match == false && player.choice < 40 && larry.choice > moe.choice)) ||

(larry.order == 0 && larry.choice < 40 && moe.choice > 39 && (player.match == true && curly.match == true && larry.choice > player.choice && larry.choice > curly.choice) || (player.match == true && curly.match == false && curly.choice < 40 && larry.choice > player.choice) || (curly.match == true && player.match == false && player.choice < 40 && larry.choice > curly.choice)) ||

(player.order == 0 && player.choice < 40 && larry.match == true && moe.match == true && curly.choice > 39 && larry.choice > player.choice && larry.choice > moe.choice) ||

(player.order == 0 && player.choice < 40 && larry.match == true && moe.match == false && moe.choice < 40 && curly.choice > 39 && larry.choice > player.choice && larry.choice > moe.choice) ||

(player.order == 0 && player.choice < 40 && larry.match == true && curly.match == true && moe.choice > 39 && larry.choice > player.choice && larry.choice > curly.choice)||

(player.order == 0 && player.choice < 40 && larry.match == true && curly.match == false && curly.choice < 40 && moe.choice > 39 && larry.choice > player.choice && larry.choice > curly.choice)||

(curly.order == 0 && curly.choice < 40 && larry.match == true && moe.match == true && player.choice > 39 && larry.choice > curly.choice && larry.choice > moe.choice) ||

(curly.order == 0 && curly.choice < 40 && larry.match == true && moe.match == false && moe.choice < 40 && player.choice > 39 && larry.choice > curly.choice && larry.choice > moe.choice) ||

(curly.order == 0 && curly.choice < 40 && larry.match == true && player.match == true && moe.choice > 39 && larry.choice > curly.choice && larry.choice > player.choice)||

(curly.order == 0 && curly.choice < 40 && larry.match == true && player.match == false && player.choice < 40 && moe.choice > 39 && larry.choice > curly.choice && larry.choice > player.choice)||

(moe.order == 0 && moe.choice < 40 && larry.match == true && curly.match == true && player.choice > 39 && larry.choice > moe.choice && larry.choice > curly.choice) ||

(moe.order == 0 && moe.choice < 40 && larry.match == true && curly.match == false && curly.choice < 40 && player.choice > 39 && larry.choice > moe.choice && larry.choice > curly.choice) ||

(moe.order == 0 && moe.choice < 40 && larry.match == true && player.match == true && curly.choice > 39 && larry.choice > moe.choice && larry.choice > player.choice)||

(moe.order == 0 && moe.choice < 40 && larry.match == true && player.match == false && player.choice < 40 && curly.choice > 39 && larry.choice > moe.choice && larry.choice > player.choice))

{

cout << "Larry takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ){

larry.tScore += 14;

player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2;

}

// No one played Q spades

else {

larry.tScore += 1;

player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2;

}

}

// Curly wins trick

else if((curly.order == 0 && curly.choice > 39 && player.choice < 40 && larry.choice < 40 && moe.choice < 40) ||

(curly.order == 0 && curly.choice < 40 && player.choice > 39 && (larry.match == true && moe.match == true && curly.choice > larry.choice && curly.choice > moe.choice) || (larry.match == true && moe.match == false && moe.choice < 40 && curly.choice > larry.choice) || (moe.match == true && larry.match == false && larry.choice < 40 && curly.choice > moe.choice)) ||

(curly.order == 0 && curly.choice < 40 && larry.choice > 39 && (player.match == true && moe.match == true && curly.choice > player.choice && curly.choice > moe.choice) || (player.match == true && moe.match == false && moe.choice < 40 && curly.choice > player.choice) || (moe.match == true && player.match == false && player.choice < 40 && curly.choice > moe.choice)) ||

(curly.order == 0 && curly.choice < 40 && moe.choice > 39 && (player.match == true && larry.match == true && curly.choice > player.choice && curly.choice > larry.choice) || (player.match == true && larry.match == false && larry.choice < 40 && curly.choice > player.choice) || (larry.match == true && player.match == false && player.choice < 40 && curly.choice > larry.choice)) ||

(player.order == 0 && player.choice < 40 && curly.match == true && moe.match == true && larry.choice > 39 && curly.choice > player.choice && curly.choice > moe.choice) ||

(player.order == 0 && player.choice < 40 && curly.match == true && moe.match == false && moe.choice < 40 && larry.choice > 39 && curly.choice > player.choice && curly.choice > moe.choice) ||

(player.order == 0 && player.choice < 40 && curly.match == true && larry.match == true && moe.choice > 39 && curly.choice > player.choice && curly.choice > larry.choice)||

(player.order == 0 && player.choice < 40 && curly.match == true && larry.match == false && larry.choice < 40 && moe.choice > 39 && curly.choice > player.choice && curly.choice > larry.choice)||

(larry.order == 0 && larry.choice < 40 && curly.match == true && moe.match == true && player.choice > 39 && curly.choice > larry.choice && curly.choice > moe.choice) ||

(larry.order == 0 && larry.choice < 40 && curly.match == true && moe.match == false && moe.choice < 40 && player.choice > 39 && curly.choice > larry.choice && curly.choice > moe.choice) ||

(larry.order == 0 && larry.choice < 40 && curly.match == true && player.match == true && moe.choice > 39 && curly.choice > larry.choice && curly.choice > player.choice)||

(larry.order == 0 && larry.choice < 40 && curly.match == true && player.match == false && player.choice < 40 && moe.choice > 39 && curly.choice > larry.choice && curly.choice > player.choice)||

(moe.order == 0 && moe.choice < 40 && curly.match == true && larry.match == true && player.choice > 39 && curly.choice > moe.choice && curly.choice > larry.choice) ||

(moe.order == 0 && moe.choice < 40 && curly.match == true && larry.match == false && larry.choice < 40 && player.choice > 39 && curly.choice > moe.choice && curly.choice > larry.choice) ||

(moe.order == 0 && moe.choice < 40 && curly.match == true && player.match == true && larry.choice > 39 && curly.choice > moe.choice && curly.choice > player.choice)||

(moe.order == 0 && moe.choice < 40 && curly.match == true && player.match == false && player.choice < 40 && larry.choice > 39 && curly.choice > moe.choice && curly.choice > player.choice))

{

cout << "Curly takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ){

curly.tScore += 14;

player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1;

}

// No one played Q spades

else {

curly.tScore += 1;

player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1;

}

}

// Moe wins trick

else if((moe.order == 0 && moe.choice > 39 && player.choice < 40 && larry.choice < 40 && curly.choice < 40) ||

(moe.order == 0 && moe.choice < 40 && player.choice > 39 && (larry.match == true && curly.match == true && moe.choice > larry.choice && moe.choice > curly.choice) || (larry.match == true && curly.match == false && curly.choice < 40 && moe.choice > larry.choice) || (curly.match == true && larry.match == false && larry.choice < 40 && moe.choice > curly.choice)) ||

(moe.order == 0 && moe.choice < 40 && larry.choice > 39 && (player.match == true && curly.match == true && moe.choice > player.choice && moe.choice > curly.choice) || (player.match == true && curly.match == false && curly.choice < 40 && moe.choice > player.choice) || (curly.match == true && player.match == false && player.choice < 40 && moe.choice > curly.choice)) ||

(moe.order == 0 && moe.choice < 40 && curly.choice > 39 && (player.match == true && larry.match == true && moe.choice > player.choice && moe.choice > larry.choice) || (player.match == true && larry.match == false && larry.choice < 40 && moe.choice > player.choice) || (larry.match == true && player.match == false && player.choice < 40 && moe.choice > larry.choice)) ||

(player.order == 0 && player.choice < 40 && moe.match == true && curly.match == true && larry.choice > 39 && moe.choice > player.choice && moe.choice > curly.choice) ||

(player.order == 0 && player.choice < 40 && moe.match == true && curly.match == false && curly.choice < 40 && larry.choice > 39 && moe.choice > player.choice && moe.choice > curly.choice) ||

(player.order == 0 && player.choice < 40 && moe.match == true && larry.match == true && curly.choice > 39 && moe.choice > player.choice && moe.choice > larry.choice)||

(player.order == 0 && player.choice < 40 && moe.match == true && larry.match == false && larry.choice < 40 && curly.choice > 39 && moe.choice > player.choice && moe.choice > larry.choice)||

(larry.order == 0 && larry.choice < 40 && moe.match == true && curly.match == true && player.choice > 39 && moe.choice > larry.choice && moe.choice > curly.choice) ||

(larry.order == 0 && larry.choice < 40 && moe.match == true && curly.match == false && curly.choice < 40 && player.choice > 39 && moe.choice > larry.choice && moe.choice > curly.choice) ||

(larry.order == 0 && larry.choice < 40 && moe.match == true && player.match == true && curly.choice > 39 && moe.choice > larry.choice && moe.choice > player.choice)||

(larry.order == 0 && larry.choice < 40 && moe.match == true && player.match == false && player.choice < 40 && curly.choice > 39 && moe.choice > larry.choice && moe.choice > player.choice)||

(curly.order == 0 && curly.choice < 40 && moe.match == true && larry.match == true && player.choice > 39 && moe.choice > curly.choice && moe.choice > larry.choice) ||

(curly.order == 0 && curly.choice < 40 && moe.match == true && larry.match == false && larry.choice < 40 && player.choice > 39 && moe.choice > curly.choice && moe.choice > larry.choice) ||

(curly.order == 0 && curly.choice < 40 && moe.match == true && player.match == true && larry.choice > 39 && moe.choice > curly.choice && moe.choice > player.choice)||

(curly.order == 0 && curly.choice < 40 && moe.match == true && player.match == false && player.choice < 40 && larry.choice > 39 && moe.choice > curly.choice && moe.choice > player.choice))

{

cout << "Moe takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37){

moe.tScore += 14;

player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0;

}

// No one played Q spades

else {

moe.tScore += 1;

player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0;

}

}

}

// No one played a heart

else if (player.choice < 40 && larry.choice < 40 && curly.choice < 40 && moe.choice < 40){

if(player.choice != 37 && larry.choice != 37 && curly.choice != 37 && moe.choice != 37) {

cout << "Trick worth no points." << endl;

}

else if(larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ) {

cout << "Trick worth thirteen points." << endl;

}

// Player wins trick

if((player.order == 0 && moe.match == true && larry.match == true && curly.match == true && player.choice > larry.choice && player.choice > curly.choice && player.choice > moe.choice) ||

(player.order == 0 && moe.match == true && larry.match == true && curly.match == false && player.choice > larry.choice && player.choice > moe.choice) ||

(player.order == 0 && curly.match == true && larry.match == true && moe.match == false && player.choice > larry.choice && player.choice > curly.choice) ||

(player.order == 0 && moe.match == true && curly.match == true && larry.match == false && player.choice > curly.choice && player.choice > moe.choice) ||

(player.order == 0 && moe.match == true && larry.match == true && curly.match == false && player.choice > larry.choice && player.choice > moe.choice) ||

(player.order == 0 && moe.match == true && larry.match == false && curly.match == false && player.choice > moe.choice) ||

(player.order == 0 && larry.match == true && moe.match == false && curly.match == false && player.choice > larry.choice) ||

(player.order == 0 && curly.match == true && larry.match == false && moe.match == false && player.choice > curly.choice) ||

(player.order == 0 && moe.match == false && larry.match == false && curly.match == false) ||

(larry.order == 0 && moe.match == true && player.match == true && curly.match == true && player.choice > larry.choice && player.choice > curly.choice && player.choice > moe.choice) ||

(larry.order == 0 && curly.match == true && player.match == true && moe.match == false && player.choice > larry.choice && player.choice > curly.choice) ||

(larry.order == 0 && moe.match == true && player.match == true && curly.match == false && player.choice > larry.choice && player.choice > moe.choice) ||

(larry.order == 0 && player.match == true && moe.match == false && curly.match == false && player.choice > larry.choice) ||

(curly.order == 0 && moe.match == true && player.match == true && larry.match == true && player.choice > curly.choice && player.choice > larry.choice && player.choice > moe.choice) ||

(curly.order == 0 && larry.match == true && player.match == true && moe.match == false && player.choice > curly.choice && player.choice > larry.choice) ||

(curly.order == 0 && moe.match == true && player.match == true && larry.match == false && player.choice > curly.choice && player.choice > moe.choice) ||

(curly.order == 0 && player.match == true && moe.match == false && larry.match == false && player.choice > curly.choice) ||

(moe.order == 0 && curly.match == true && player.match == true && larry.match == true && player.choice > moe.choice && player.choice > larry.choice && player.choice > curly.choice) ||

(moe.order == 0 && larry.match == true && player.match == true && curly.match == false && player.choice > moe.choice && player.choice > larry.choice) ||

(moe.order == 0 && curly.match == true && player.match == true && larry.match == false && player.choice > moe.choice && player.choice > curly.choice) ||

(moe.order == 0 && player.match == true && curly.match == false && larry.match == false && player.choice > moe.choice)){

cout << "Player takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37){

player.tScore += 13;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

// No one played Q spades

else {

player.tScore += 0;

player.order = 0; larry.order = 1; curly.order = 2; moe.order = 3;

}

}

// Larry wins trick

else if((larry.order == 0 && moe.match == true && player.match == true && curly.match == true && larry.choice > player.choice && larry.choice > curly.choice && larry.choice > moe.choice) ||

(larry.order == 0 && moe.match == true && player.match == true && curly.match == false && larry.choice > player.choice && larry.choice > moe.choice) ||

(larry.order == 0 && curly.match == true && player.match == true && moe.match == false && larry.choice > player.choice && larry.choice > curly.choice) ||

(larry.order == 0 && moe.match == true && curly.match == true && player.match == false && larry.choice > curly.choice && larry.choice > moe.choice) ||

(larry.order == 0 && moe.match == true && player.match == true && curly.match == false && larry.choice > player.choice && larry.choice > moe.choice) ||

(larry.order == 0 && moe.match == true && player.match == false && curly.match == false && larry.choice > moe.choice) ||

(larry.order == 0 && player.match == true && moe.match == false && curly.match == false && larry.choice > player.choice) ||

(larry.order == 0 && curly.match == true && player.match == false && moe.match == false && larry.choice > curly.choice) ||

(larry.order == 0 && moe.match == false && player.match == false && curly.match == false) ||

(player.order == 0 && moe.match == true && larry.match == true && curly.match == true && larry.choice > player.choice && larry.choice > curly.choice && larry.choice > moe.choice) ||

(player.order == 0 && curly.match == true && larry.match == true && moe.match == false && larry.choice > player.choice && larry.choice > curly.choice) ||

(player.order == 0 && moe.match == true && larry.match == true && curly.match == false && larry.choice > player.choice && larry.choice > moe.choice) ||

(player.order == 0 && larry.match == true && moe.match == false && curly.match == false && larry.choice > player.choice) ||

(curly.order == 0 && moe.match == true && larry.match == true && player.match == true && larry.choice > curly.choice && larry.choice > player.choice && larry.choice > moe.choice) ||

(curly.order == 0 && player.match == true && larry.match == true && moe.match == false && larry.choice > curly.choice && larry.choice > player.choice) ||

(curly.order == 0 && moe.match == true && larry.match == true && player.match == false && larry.choice > curly.choice && larry.choice > moe.choice) ||

(curly.order == 0 && larry.match == true && moe.match == false && player.match == false && larry.choice > curly.choice) ||

(moe.order == 0 && curly.match == true && larry.match == true && player.match == true && larry.choice > moe.choice && larry.choice > player.choice && larry.choice > curly.choice) ||

(moe.order == 0 && player.match == true && larry.match == true && curly.match == false && larry.choice > moe.choice && larry.choice > player.choice) ||

(moe.order == 0 && curly.match == true && larry.match == true && player.match == false && larry.choice > moe.choice && larry.choice > curly.choice) ||

(moe.order == 0 && larry.match == true && curly.match == false && player.match == false && larry.choice > moe.choice)) {

cout << "Larry takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ){

larry.tScore += 13;

player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2;

}

// No one played Q spades

else {

larry.tScore += 0;

player.order = 3; larry.order = 0; curly.order = 1; moe.order = 2;

}

}

// Curly wins trick

else if((curly.order == 0 && moe.match == true && player.match == true && larry.match == true && curly.choice > player.choice && curly.choice > larry.choice && curly.choice > moe.choice) ||

(curly.order == 0 && moe.match == true && player.match == true && larry.match == false && curly.choice > player.choice && curly.choice > moe.choice) ||

(curly.order == 0 && larry.match == true && player.match == true && moe.match == false && curly.choice > player.choice && curly.choice > larry.choice) ||

(curly.order == 0 && moe.match == true && larry.match == true && player.match == false && curly.choice > larry.choice && curly.choice > moe.choice) ||

(curly.order == 0 && moe.match == true && player.match == true && larry.match == false && curly.choice > player.choice && curly.choice > moe.choice) ||

(curly.order == 0 && moe.match == true && player.match == false && larry.match == false && curly.choice > moe.choice) ||

(curly.order == 0 && player.match == true && moe.match == false && larry.match == false && curly.choice > player.choice) ||

(curly.order == 0 && larry.match == true && player.match == false && moe.match == false && curly.choice > larry.choice) ||

(curly.order == 0 && moe.match == false && player.match == false && larry.match == false) ||

(player.order == 0 && moe.match == true && curly.match == true && larry.match == true && curly.choice > player.choice && curly.choice > larry.choice && curly.choice > moe.choice) ||

(player.order == 0 && larry.match == true && curly.match == true && moe.match == false && curly.choice > player.choice && curly.choice > larry.choice) ||

(player.order == 0 && moe.match == true && curly.match == true && larry.match == false && curly.choice > player.choice && curly.choice > moe.choice) ||

(player.order == 0 && curly.match == true && moe.match == false && larry.match == false && curly.choice > player.choice) ||

(larry.order == 0 && moe.match == true && curly.match == true && player.match == true && curly.choice > larry.choice && curly.choice > player.choice && curly.choice > moe.choice) ||

(larry.order == 0 && player.match == true && curly.match == true && moe.match == false && curly.choice > larry.choice && curly.choice > player.choice) ||

(larry.order == 0 && moe.match == true && curly.match == true && player.match == false && curly.choice > larry.choice && curly.choice > moe.choice) ||

(larry.order == 0 && curly.match == true && moe.match == false && player.match == false && curly.choice > larry.choice) ||

(moe.order == 0 && larry.match == true && curly.match == true && player.match == true && curly.choice > moe.choice && curly.choice > player.choice && curly.choice > larry.choice) ||

(moe.order == 0 && player.match == true && curly.match == true && larry.match == false && curly.choice > moe.choice && curly.choice > player.choice) ||

(moe.order == 0 && larry.match == true && curly.match == true && player.match == false && curly.choice > moe.choice && curly.choice > larry.choice) ||

(moe.order == 0 && curly.match == true && larry.match == false && player.match == false && curly.choice > moe.choice)) {

cout << "Curly takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37 ){

curly.tScore += 13;

player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1;

}

// No one played Q spades

else {

curly.tScore += 0;

player.order = 2; larry.order = 3; curly.order = 0; moe.order = 1;

}

}

// Moe wins trick

else if((moe.order == 0 && curly.match == true && player.match == true && larry.match == true && moe.choice > player.choice && moe.choice > larry.choice && moe.choice > curly.choice) ||

(moe.order == 0 && curly.match == true && player.match == true && larry.match == false && moe.choice > player.choice && moe.choice > curly.choice) ||

(moe.order == 0 && larry.match == true && player.match == true && curly.match == false && moe.choice > player.choice && moe.choice > larry.choice) ||

(moe.order == 0 && curly.match == true && larry.match == true && player.match == false && moe.choice > larry.choice && moe.choice > curly.choice) ||

(moe.order == 0 && curly.match == true && player.match == true && larry.match == false && moe.choice > player.choice && moe.choice > curly.choice) ||

(moe.order == 0 && curly.match == true && player.match == false && larry.match == false && moe.choice > curly.choice) ||

(moe.order == 0 && player.match == true && curly.match == false && larry.match == false && moe.choice > player.choice) ||

(moe.order == 0 && larry.match == true && player.match == false && curly.match == false && moe.choice > larry.choice) ||

(moe.order == 0 && curly.match == false && player.match == false && larry.match == false) ||

(player.order == 0 && curly.match == true && moe.match == true && larry.match == true && moe.choice > player.choice && moe.choice > larry.choice && moe.choice > curly.choice) ||

(player.order == 0 && larry.match == true && moe.match == true && curly.match == false && moe.choice > player.choice && moe.choice > larry.choice) ||

(player.order == 0 && curly.match == true && moe.match == true && larry.match == false && moe.choice > player.choice && moe.choice > curly.choice) ||

(player.order == 0 && moe.match == true && curly.match == false && larry.match == false && moe.choice > player.choice) ||

(larry.order == 0 && curly.match == true && moe.match == true && player.match == true && moe.choice > larry.choice && moe.choice > player.choice && moe.choice > curly.choice) ||

(larry.order == 0 && player.match == true && moe.match == true && curly.match == false && moe.choice > larry.choice && moe.choice > player.choice) ||

(larry.order == 0 && curly.match == true && moe.match == true && player.match == false && moe.choice > larry.choice && moe.choice > curly.choice) ||

(larry.order == 0 && moe.match == true && curly.match == false && player.match == false && moe.choice > larry.choice) ||

(curly.order == 0 && larry.match == true && moe.match == true && player.match == true && moe.choice > curly.choice && moe.choice > player.choice && moe.choice > larry.choice) ||

(curly.order == 0 && player.match == true && moe.match == true && larry.match == false && moe.choice > curly.choice && moe.choice > player.choice) ||

(curly.order == 0 && larry.match == true && moe.match == true && player.match == false && moe.choice > curly.choice && moe.choice > larry.choice) ||

(curly.order == 0 && moe.match == true && larry.match == false && player.match == false && moe.choice > curly.choice)) {

cout << "Moe takes the trick." << endl;

// Check for Q spades

if (larry.choice == 37 || curly.choice == 37 || moe.choice == 37 || player.choice == 37){

moe.tScore += 13;

player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0;

}

// No one played Q spades

else {

moe.tScore += 0;

player.order = 1; larry.order = 2; curly.order = 3; moe.order = 0;

}

}

}

cout << "Player hand is " << player.tScore << "\t"

<< "Larry hand is " << larry.tScore << "\t\t"

<< "Curly hand is " << curly.tScore << "\t\t"

<< "Moe hand is " << moe.tScore << endl;

}

void count() {

static int counter = 0;

counter++;

// Reset count to 0 when it hits 13

if (counter == 13) {

counter = 0; // Reset count to 0

}

}