Project 2

Title:

**21 the card game**

Due Date:

July 31, 2015

Author:

Hector Espinoza

**Title:**

21 the card game

**Introduction:**

21 is played with an international set of 52 cards. Each player is dealed two cards on the first round. When the game begins the player looks at their cards and o hold two or more cards which together sum to 21 (ideally an Ace and a tenth, this is known as pontoon or a natural) but above 15 and as near to 21 as possible. Any hand that exceeds 21 is “bust” and the player loses that round.

**How to play:**

Each player has an independent game against the dealer and the objective is to beat the dealer. If the player's cards total is higher than 21, it is called a "bust" and the dealer wins, even if the dealer have a bust as well and if the dealer and the player have the same value it’s a “bust”. A 21 hand value is calculated as follows: Cards of 2 - 10 are worth their face value. Jack, Queen and King are also worth 10 each. The Ace card is worth 11, unless it causes the player to bust his hand and in this case the Ace card can be counted as 1. A 2 cards hand with ace and a 10 value card is called a "blackjack" and it is a winning hand unless the dealer have an ace and a 10 value 2 cards hand is well, in this case it is a “bust”.

**The game course:**

After all players have placed their bets, the dealer deals each player with 2 cards including himself, the dealers first card will be face up and the other is face down.

21 player play options:

1. Hit - Get another card.

2. Stand - Stay with the current hand..

After all players have played their hand, the dealer revels his other card and start to play his hand.

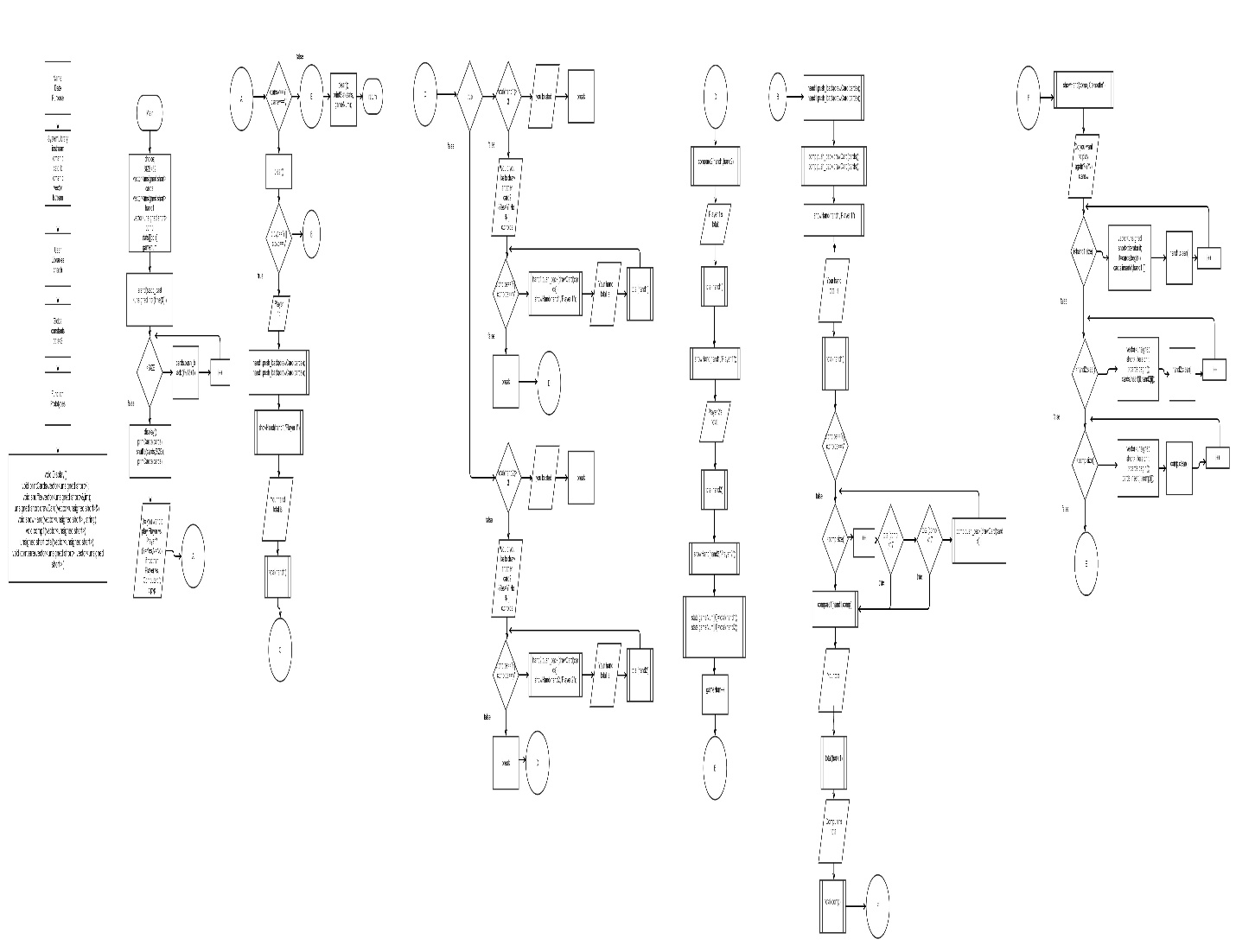
House rules say that the dealer must continue to hit until he has at least 17.

**Summary:**

**Project Size:** 400+ lines

I tried to make this game simple and easy to use, my main vision for this game was to be based off of simplicity so that the player will have fun playing and can play for perhaps hours. I think that I made it work really well, I so far am addicted to it and want to keep playing it over and over. A way that I thought would be best so that the game would be fast and fun was to be player versus computer. I believe that I worked it out really well and made the game a fair game so that the computer and player felt that it wouldn’t be one sided. Another thing I tried to implement was the use of only 4 of each number, I could have set a random number generator and get any number a multiple of times. Just in a real life game you cannot pull a card more than 4 times and I was fortunate enough to use an array and be able to put that restriction on the game.

**Flow chart:**



**Pseudo code:**

Random generator, generates numbers from 1 to 13 to get all the cards of the 52 deck (without jokers).

Displays the rules of the game.

For loops to the size and does them all from 1 to 13 and at each row it will create a new row once it reaches 13.

Calls the shuffle function.

Uses a for loop and random number generator then swaps the positions in the array.

Calls the print card function.

Ask the user if he wants to play player vs. player.

If the answer is yes then it calls the clear function to clear the board.

Calls the draw card function twice and then the show hand function for player 1.

Uses an if statement to determine in your hand is a bust, if so then it automatically says the other person wins and the game is over. If not then the player may draw another card.

Until the player says he does not wish to draw more cards is when the player 1’s turn is over.

Calls the clear screen function.

Calls the draw card function twice and then the show hand function for player 2.

Uses an if statement to determine in your hand is a bust, if so then it automatically says the other person wins and the game is over. If not then the player may draw another card.

Until the player says he does not wish to draw more cards is when the player 2’s turn is over.

Calls the clear screen function.

Then it calls the compare2 function to compare player 1’s hand vs. player 2’s hand.

It will output the winner and player 1’s hand total and the hand itself and same for player 2.

At the end it keeps track of the total score for each game played.

Once the players both do not want to play again it will display the stats of all the games.

If the player wants to play player vs. computer.

It draws two cards and Calls the show hand function.

Uses an if statement to determine in your hand is a bust, if so then it automatically says the other person wins and the game is over. If not then the player may draw another card.

While the player is determining if he should draw another card the computer is also drawing cards as long as his total is less than 17.

While if the answer is anything other than yes it will loop and to the size of the array and will to determine if the computer needs to draw another card or not.

Calls the compare function.

If player is greater than computer and is less than 21 then the player wins. If the player is equal to 21 the player wins. If the player is less than 21 and computer is greater than 21 the player wins. Any other condition the player loses.

The next step couts the players total hand.

Calls the total function for player 1.

Uses for loops from 2 to 10 and sums them up for their face value, jack queen and king have a value of 10 and the Ace has a value of 1 or 11 based on if it passes the max limit of 21.

Then couts the computers total hand.

Uses for loops from 2 to 10 and sums them up for their face value, jack queen and king have a value of 10 and the Ace has a value of 1 or 11 based on if it passes the max limit of 21.

Then finally will display the computers hand.

Calls the show hand function for the computer.

At the end of every game the players hand will go back in the deck on the opposite end, allowing the player to count cards if he wishes, but caution don’t get caught!

**Major Variables**

|  |  |  |
| --- | --- | --- |
| Function | Decscription | location |
| Cin/cout | Used to input and output information | Throughout the code |
| Local scope | The sum of the total of cards | In function unsigned short total |
| Global scope | Const int SIZE | In main |
| char | Choice to ask the user if he wants to draw another card | In main |
| short | Used to set the size of the vectors | In main the vectors |
| string | Computer name and player name | In show hand function |
| comments | Used to comment the code | Throughout the code |
| If ,If-else | To determine a condition | In the function total |
| While,do-while | To determine a condition to be true or false, and will loop till it is true. | In main |
| for | Is a counter for what in the array | In the show hand function |
| protoypying | Making functions | Before main |
| Void vs. return | Used to display rules | After main |
| Random number generator | Generates random numbers | In print cards function |
| array | Creates a table of information | In print cards |
| vector | Used to call the functions in the library | In function shuffle |
| Pop\_back | Push the cards back and deletes the last one | Draw card function |
| Push\_back | Pushed the desired slot to the end | In main |
| Increment and decrement | Counts up 1 or down 1 | In the show hand function in the for loop |
| Set width | Uses to set the desired spacing | In print cards function |
| casting | Re type a variable | In main, after declare variables |
| Arithmetic operators | Used to determine which hand was greater the player or the computer | In the function compare |
| Open file | Used to display the rules of the game | In void display |
| 2 dimensional array | Used to hold the stats of the games | In print stats function |
| Iterator | Learned basic iterator outside of what was taught in class to put the cards back into the deck | At the end of main |
| Structure | Used structure to hold the responses | In the header file |
| Module | Used module to only allow 51 cards in order to make a deck | After declaring my variables |

Refrence:

1. Cplusplus the online resource( used as a reference for vectors)
2. The book Gaddis 11th Edition

Code:

//System Libraries

#include <iostream>

#include <string>

#include <cstdlib>

#include <iomanip>

#include <vector>

#include <fstream>

using namespace std;

//User Libraries

#include "char.h"

//Global Constants

const int COLS=2;

//Function Prototypes

void Display(); //Displays the rules of the game

void clear(); //Clears the screen for the new game

void printCards(vector<unsigned short>); //Prints cards 4 of each

void shuffle(vector<unsigned short>&,int); //Shuffles the cards

void showHand(vector<unsigned short> , string); //Shows the players hand

void comp1(vector<unsigned short>); //Computers hand of cards

void compare1(vector<unsigned short> ,vector<unsigned short> ); //Compares the computer and the users cards to determine the winner

void compare2(vector<unsigned short> ,vector<unsigned short> ); //Compares the player and the users cards to determine the winner

unsigned short total(vector<unsigned short>); //Total of the players card

unsigned short drawCard(vector<unsigned short>&); //Draws one card at a time from the bottom of the deck

void printStat(int[][COLS],int);

//Execution Begins Here!

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

//Declare Variables

const int SIZE=52; //Size of the array

choices c; //The choices in the structure

vector <unsigned short> cards; //Vector for the cards

vector <unsigned short> hand1; //Holds player 1's

vector <unsigned short> hand2; //Holds player 2's hand

vector <unsigned short> comp; //Holds the computers hand cards

int stats[10][COLS];

int gameNum=0;

//Vector for the players hand to store what cards he has

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

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

cards.push\_back (i%13+1);

}

Display();

printCards(cards);

shuffle(cards,SIZE);

printCards(cards);

//Ask the user if they want to play player vs. player or computer

cout<<"Do you want to play Player vs. Player?(Y=Yes,N=No) if not then Player vs. Computer: ";

cin>>c.pvp;

do{

clear();

if(c.pvp=='Y'||c.pvp=='y'){

//Draws two cards

cout<<"Player 1: ";

hand1.push\_back(drawCard(cards));

hand1.push\_back(drawCard(cards));

showHand(hand1,"Player 1");

cout<<"Your hand total is "<<total(hand1)<<endl;

while(true){

if(total(hand1)>21){

cout<<"You busted"<<endl;

break;

}

cout<<"Would you like to draw another card?(Yes='Y',No='N') ";

cin>>c.choice;

if (c.choice=='Y'||c.choice=='y'){

hand1.push\_back(drawCard(cards));

showHand(hand1,"Player 1");

cout<<"Your hand total is "<<total(hand1);

}else{

break;

}

cout<<endl;

}

clear();

cout<<"Player 2: ";

//Draw two cards

hand2.push\_back(drawCard(cards));

hand2.push\_back(drawCard(cards));

//Shows the players hand

showHand(hand2,"Player 2");

cout<<"Your hand total is "<<total(hand2)<<endl;

while(true){

if(total(hand1)>21){

cout<<"You busted"<<endl;

break;

}

cout<<"Would you like to draw another card?(Yes='Y',No='N') ";

cin>>c.choice;

if (c.choice=='Y'||c.choice=='y'){

hand2.push\_back(drawCard(cards));

showHand(hand2,"Player 2");

cout<<"Your hand total is "<<total(hand2);

}else{

break;

}

cout<<endl;

}

compare2(hand1,hand2);

cout<<endl;

cout<<"Player 1's total: "<<total(hand1)<<endl;

showHand(hand1,"Player 1");

cout<<"Player 2's total: "<<total(hand2)<<endl;

showHand(hand2,"Player 2");

cout<<endl;

//Keeps track of the players scores

stats[gameNum][0]=total(hand1);

stats[gameNum][1]=total(hand2);

gameNum++;

}else{

//Plays computer vs. Player

//Draws the players cards

hand1.push\_back(drawCard(cards));

hand1.push\_back(drawCard(cards));

//Draws the computers cards

comp.push\_back(drawCard(cards));

comp.push\_back(drawCard(cards));

//Shows the players cards

showHand(hand1,"Player 1");

cout<<"Your hand total is "<<total(hand1)<<endl;

do{

if(total(hand1)>21){

cout<<"You busted"<<endl;

break;

}

cout<<"Would you like to draw another card?(Yes='Y',No='N') ";

cin>>c.choice;

//Used to draw the players cards and the computers card

if (c.choice=='Y'||c.choice=='y'){

hand1.push\_back(drawCard(cards));

if(total(comp)<17){

comp.push\_back(drawCard(cards));

}

showHand(hand1,"Player 1");

cout<<"Your hand total is "<<total(hand1);

}

cout<<endl;

//Used to for-loop until the computer no longer needs cards

}while(c.choice=='Y'||c.choice=='y');

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

do{

if(total(comp)<17){

comp.push\_back(drawCard(cards));

}

}while(total(comp)<17);

compare1(hand1,comp);

cout<<"Your total "<<total(hand1)<<"\n";

cout<<"Computer's total "<<total(comp)<<"\n";

showHand(comp,"Computer");

cout<<endl;

}

//Ask the user if he/she wants to play again

cout<<"Do you want to play again?(Y/N) ";

cin>>c.answ;

for(int i=0;i<hand1.size();i++){

vector<unsigned short>::iterator it;

it=cards.begin();

cards.insert(it,hand1[i]);

}

hand1.clear();

for(int i=0;i<hand2.size();i++){

vector<unsigned short>::iterator it;

it=cards.begin();

cards.insert(it,hand2[i]);

}

hand2.clear();

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

vector<unsigned short>::iterator it;

it=cards.begin();

cards.insert(it,comp[i]);

}

comp.clear();

}while(c.answ=='Y'||c.answ=='y');

clear();

printStat(stats,gameNum);

//Exit Stage Right!

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PrintStat

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose:Prints the stats for the player vs. player game

\* Input:

\* stats-> a 2 dimensinal array with row and colums.

\* N-> The hand for player 1 and player 2

\* Output->

\* The stats for the game, what game and the hand total for both players

\*/

void printStat(int stats[][COLS],int n){

cout<<"Stats for all the games played."<<endl;

cout<<"Game-Num P1 P2"<<endl;

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

cout<<setw(5)<<i+1<<setw(9)<<stats[i][0]<<" "<<setw(5)<<stats[i][1]<<"\n";

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Clear

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose:To clear the screen for the new game

\* output: 25 endl's

\*/

void clear(){

for(int i=1;i<=50;i++){

cout<<endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Total

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose:To calculate the total of the players card

\* Input:

\* vector[cards]->The cards in their hand

\* Output:

\* Sum->The sum of the cards

\*/

unsigned short total(vector<unsigned short>cards){

unsigned short sum=0;

unsigned int ace;

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

if(cards[i]==1){

if(sum<10){

sum+=11;

}

else{

sum+=1;

}

}else if(cards[i]==2){

sum+=2;

}else if(cards[i]==3){

sum+=3;

}else if(cards[i]==4){

sum+=4;

}else if(cards[i]==5){

sum+=5;

}else if(cards[i]==6){

sum+=6;

}else if(cards[i]==7){

sum+=7;

}else if(cards[i]==8){

sum+=8;

}else if(cards[i]==9){

sum+=9;

}else if(cards[i]==10){

sum+=10;

}else if(cards[i]==11){

sum+=10;

}else if(cards[i]==12){

sum+=10;

}else if(cards[i]==13){

sum+=10;

}

}

return sum;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Showhand

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose:To calculate the total of the players card

\* Input:

\* vector[cards]->The cards in their hand

\* Output:

\* Sum->The sum of the cards

\*/

void showHand(vector<unsigned short> v, string name){

cout<<name<<"'s hand\n";

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

if(v[i]==13){

cout<<"K ";

}else if(v[i]==12){

cout<<"Q ";

}else if(v[i]==11){

cout<<"J ";

}else if(v[i]==1){

cout<<"A ";

}else cout<<v[i]<<" ";

}

cout<<"\n";

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* compare

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To compare player1's hand vs. the computer's hand

\* Input:

\* vector[p]->for player1's hand

\* vector[c]->for the computer's hand

\* Output:

\* Who wins the computer or the player

\*/

void compare1(vector<unsigned short> p,vector<unsigned short> c){

if(total(p)>total(c) && total(p)<=21){

cout<<"You Win!!!!"<<endl;

}

else if(total(p)==21 && total(c)==21){

cout<<"You lose"<<endl;

}

else if(total(p)==21 && total(p)<=21){

cout<<"You Win!!!"<<endl;

}

else if(total(p)<=21 && total(c)>21){

cout<<"You win!!!"<<endl;

}

else{

cout<<"You Lose!!!"<<endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* compare2

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To compare player1's hand vs. player2's hand

\* Input:

\* vector[p1]->for player1's hand

\* vector[p2]->for player2's hand

\* Output:

\* Who wins player1 or the player2

\*/

void compare2(vector<unsigned short> p1,vector<unsigned short> p2){

if(total(p1)>total(p2) && total(p1)<=21){

cout<<"Player 1 Wins!!"<<endl;

}else if(total(p2)>total(p1) && total(p2)<=21){

cout<<"Player 2 Wins!!"<<endl;

}else if(total(p1)<=21 && total(p2)>21){

cout<<"Player 1 Wins!!!"<<endl;

}else if(total(p2)<=21 && total(p1)>21){

cout<<"Player 2 Wins!!!"<<endl;

}else if(total(p1)==21 && total(p2)==21){

cout<<"Tie"<<endl;

}else if(total(p2)==21 && total(p1)==21){

cout<<"Tie"<<endl;

}else if(total(p2)==total(p1)){

cout<<"Tie"<<endl;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PrintCards

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: Print the deck of cards

\* Input:

\* vector[v]->the list of cards

\* Output:

\* The deck of cards

\*/

void printCards(vector<unsigned short> v){

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

if(i%13==0);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Shuffle

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: To shuffle the deck

\* Input:

\* vector[&v]->A copy of the deck

\* s-> size of the array

\* Output:

\* A shuffled deck

\*/

void shuffle(vector<unsigned short> &v,int s){

for(int i=0 ; i< s\*3 ; i++){

int pos1 = rand()%s;

int pos2 = rand()%s;

unsigned short temp = v[pos1];

v[pos1]=v[pos2];

v[pos2]=temp;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* drawCard

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: Draws one card from the deck

\* Input:

\* vector[&c]->A copy of the deck

\* Output:

\* returns the temp card from the deck

\*/

unsigned short drawCard(vector<unsigned short> &c){

unsigned short temp=c[c.size()-1];

c.pop\_back();

return temp;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Display

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Purpose: Displays the rules

\* Output:

\* Outputs the rules to the game

\*/

void Display(){

//Open the file

ifstream input("display.txt");

string rules;

if(input.is\_open()){

while(getline(input,rules)){

cout<<rules<<endl;

}

}else{

cout<<"ERROR: Cannot open file."<<endl;

}

cout<<endl;

}