Project 1

<Poker>

CSC-5 43952
Branden Hitt
Date: 5/15/2015

Table of Contents

Introduction

Title: Poker (Straight Poker/Traditional Poker Variation)

This is a small program for four-handed poker (1 user and 3 ai).

At the beginning of the program, the user will be prompted to choose if they want the rules displayed or not. The rules also include the hand rankings which you can find below. From there they are asked to give their name. Following that, the program deals out a hand of 5 cards to each of the four players. The user is told his hand and its value (pair, high card, flush, etc). Next, the program will display all of the players' hands and the values. Finally, the program will display who won the hand and add a point to that person's overall score. The user may play additional rounds of poker as many times as they want. At the very end of the program, the overall scores are tallied and the overall winner is displayed.

Ex:
```
“Congrats to Phil Ivey for being the table's big winner!
 Out of 6 total rounds, Phil Ivey won 3 rounds of Poker.
 Thank you for playing!”
```

The final results for the player are then sent to a file (Results.dat)

Ex:
```
“Player Name: Branden
 Round Wins : 2
 Total Rounds: 6
 Percent Win: %33.33”
```

Hand Rankings

(Low)

High Card: This hand has nothing to rely on but a High card.

Ex: 3 Diamonds | Jack Clubs | 8 Spades | 4 Hearts | 2 Spades

Pair: Two cards of the same rank.

Ex: Ace Hearts | Ace Diamonds | 8 Clubs | 4 Spades | 7 Hearts

Three of a Kind: Three cards of the same rank.

Ex: 7 Clubs | 7 Diamonds | 7 Spades | King Clubs | 3 Diamonds

Straight: Five cards in a sequence, but not of the same suit.

Ex: 5 Hearts | 6 Diamonds | 7 Spades | 8 Diamonds | 9 Clubs

Flush: Any five cards of the same suit, but not in a sequence.

Ex: 4 Spades | Jack Spades | 8 Spades | 2 Spades | 9 Spades

Full House: Three of a Kind and a Pair.

Ex: 10 Hearts | 10 Diamonds | 10 Spades | 9 Clubs | 9 Diamonds

Four of a Kind: All four cards of the same rank.

Ex: Jack Hearts | Jack Spades | Jack Diamonds | Jack Clubs | 7 Hearts

Straight Flush: Five cards in a sequence, all in the same suit.

Ex: 4 Spades | 5 Spades | 6 Spades | 7 Spades | 8 Spades

Royal Flush: 10, J, Q, K, A (all in the same suit).

Ex: 10 Spades | Jack Spades | Queen Spades | King Spades | Ace Spades

(High)

In the event of a tie hand (ex. both players have a pair), the higher pair is chosen.

Summary

Project size: 610 lines (with comments)

Number of Variables: 71

Number of Functions: 10 (+1 for plus main)

I tried to include everything we have covered so far, and I believe I fit it all in the program except for maybe abs() function since it wasn't really needed. Functions were extremely useful for saving time and also for organization. The most useful thing might have been if()else() statements, since there ended up being a lot of logic for the program. I should also mention that even though we just covered arrays, I found using one to be extremely useful for creating the deck of cards in the first place (even though I messed up the array for awhile without noticing, off-by-one rookie error).

I am mostly satisfied with how the program came out, except for a few small adjustments that would be nice. So far, I can compare hands with the same thing (like a pair) and then give the win to the one with the higher pair.

```
Ex: 4 spades | 4hearts | 5 hearts | 8 clubs | Ace diamonds
    5 spades | 5clubs   | 7 clubs  | 8 hearts | King diamonds    ← winner with higher pair
```

The problem is that I haven't accounted for if they both have the same pair, which would mean that they need another high card .

```
Ex:  5 spades | 5clubs   | 7 clubs  | 8 hearts | King diamonds  ← should be winner with King high
     5 hearts   | 5diamonds  | 3 clubs  | 9 hearts | Jack Spades
Result: Tie game ( no winner)
```

Even though this outcome or similar ones are more rare, I would prefer to account for it happening.

There were also many things that looking back could be organized/coded to be more efficient.

Future Improvements

There are many things I would love to add to this game to improve it. Of course, having sound and graphics for the game would be nice, but that is a bit out of my reach as I am new to programming. In addition, I have already talked a little about the small chances of rare hands that I do not account for that could happen. I need to implement those fixes to find the correct winners (ie second and third high cards). Another portion of the program that I would like to add (maybe for the next project) would be the betting. Poker is much much more fun when you are betting on your hand and winning money. This would of course mean that I would need to include some sort of ai for the computer players' bets which would most likely be very simple ai or just random ai bets. These were both originally planned for this project, however it was too much for me to add in the time frame and I was having problems figuring out how it would work anyway. Lastly, I also originally wanted this game of Poker to be the Texas Hold'em variation which is a lot more complicated. Instead of 1 round of betting, there would be 4 minimum, and the way your hand is ranked would be a lot more complicated since there are community cards in the center that help make up every hand. Still, this would be a fun improvement if I could accomplish it since I thoroughly enjoy Texas Hold'em and prefer it over straight poker.

Simple Program Function Overview

```
main(
        iniDeck();  ← create 52 card deck
        info(); ← display rules/hand rankings if user wants
        round(); ← start a new round
        iniDeck(); ← shuffle deck for new round
)
round(
        deal1();  ← this is done for all 20 cards (5 per player)
        hiCard(); ← sorts the cards for hi card and ordered display
        ranking(); ← find hand rankings number for all players
        aiRank(); ← gives string for output of hand rank (ex: Pair)
        carVal(); ← gives string for individual card/ used for display to player (ex: 9 hearts)
        winHand(); ← find best hand
        winner(); ← find string for winner name
)
```

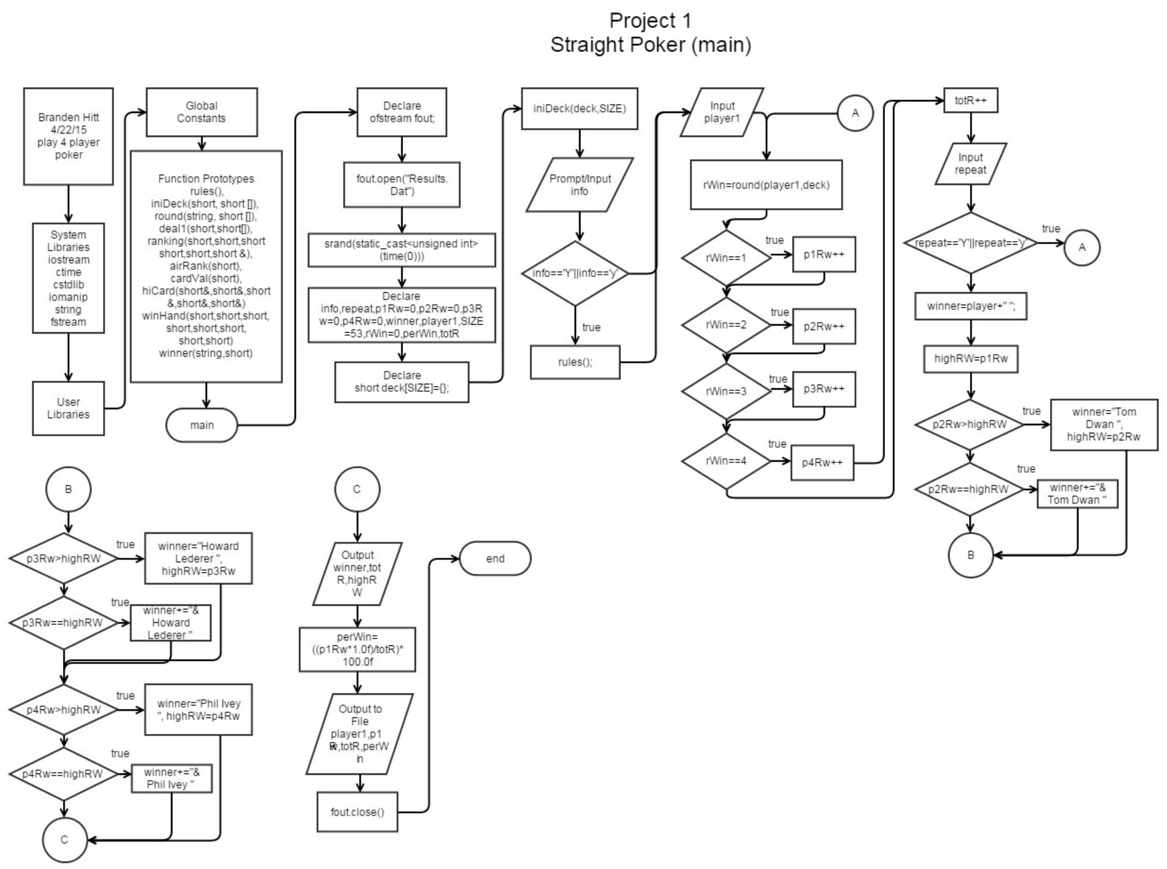
Psuedocode

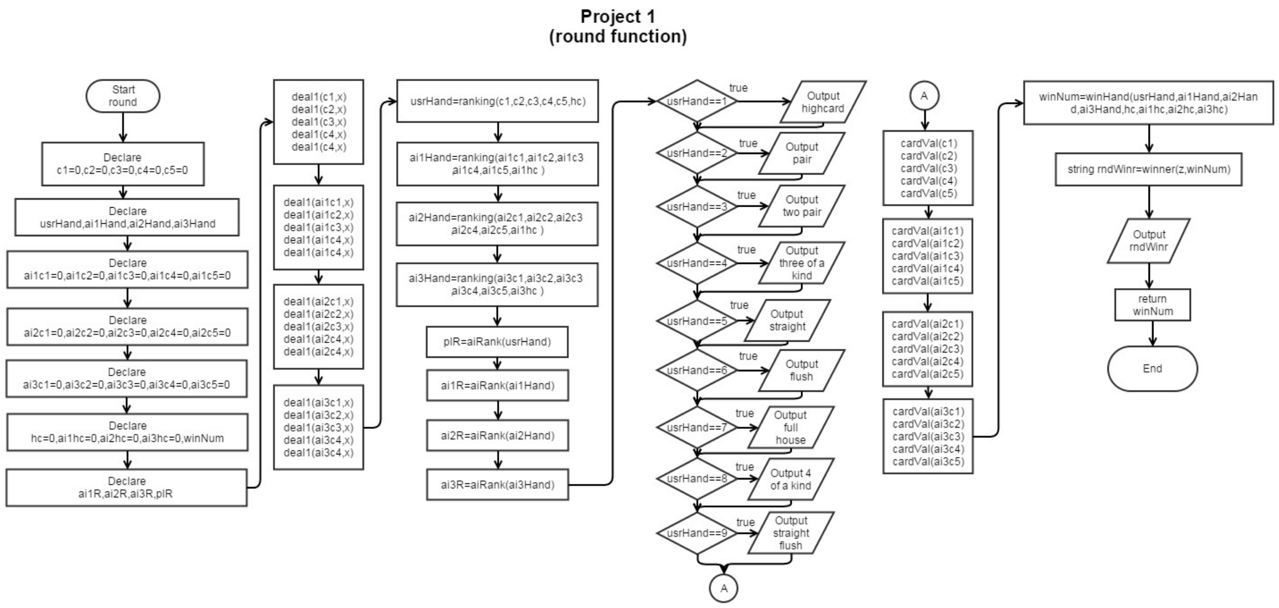
```
/*
 * File:   main.cpp
 * Author: Branden Hitt
 * Created on May 6, 2015, 1:02 PM
 *      Purpose: Code a game of straight poker
 */

//System Libraries
//I/O standard
//computer time
//for random seed
//formatting
//for strings
//file in/out

//User Libraries

//Global Constants

//Function Prototypes
//instructions on how to play
//initializes and shuffles the deck
//starts a new round of poker
//deal 1 card
//gives hand value
//finds ranking of computer hand
//finds what the card is for output
//sort for high card
//find winning hand
//find round winner

//Execution Begins Here!

    //Declare Variables
    //output to file
    //open file
    //set random seed
    //input for info query, and repeat query
    //total wins for each player
    //overall winner and player name input
    //size of the deck
    //declare array with zeros
    //current round win
    //percentage of wins for the player/ totals rounds
    //fill array with card values (1-52)
    //Prompt for rules
    //prompt for name of player
    //input player name
    //pause
    //do
        //Start new round
        //call round function
        //if 1 player 1 win round increment their win
        //if 2 player 2 win round increment their win
        //if 3 player 3 win round increment their win
        //if 4 player 4 win round increment their win
        //Shuffle
        //increment the total rounds
        //prompt for new round
    //while player wants to repeat
    //Congratulate Overall Winner
    //find percent win
    //output results to a file

    //Exit Stage Right!

//*********************************************//
//*            Rules Function               *//
//*********************************************//
    //declare variables
    //menu / choice
    //bool for skip
    //begin explanation of how to play

    //do
        //start showing hand rankings
        //prompt for specific hand ranking display
        //menu for hand display
            //case for hi card
            //case for pair
            //case for two pair
            //case for three of a kind
            //case for straight
            //case for flush
            //case for full house
            //case for four of a kind
            //case for straight flush
            //case for royal flush
            //default for skip
                //cho = n
                //set skip to true
        //if skip is false
        //prompt for repeat
//while player wants to repeat
//*********************************************//
//*            Initialize Deck             *//
//*********************************************//
//increment while still in array
        //for position i, set i to i ex: 41 =41
//*********************************************//
//*              New round                 *//
//*********************************************//
    //Declare variables
    //player cards
    //hand values
    //computer 1 hand
    //computer 2 hand
    //computer 3 hand
    //high card for tie hands
    //player number who has top hand
    //quick string for hand ranking
    //Deal individual hands
    //high card and sort
    //find hand value
    //set string for ai hand ranking
    //state player hand value
    //Display player hand
    //Unveil all hands
    //find winning hand
    //find winNum
    //find winner name
    //state winner
    //return winner number
//*********************************************//
//*              Deal 1                      *//
//*********************************************//
    //declare variables
    //repeat
    //new card
    //do
        //pick random number for card
        //if card is 0, then its already picked, so repeat
        //else dont repeat
    //while need new card
    //set the card you picked to zero in array
    //c1 = card
//*********************************************//
//*            Card Values                   *//
//*********************************************//
    //declare string f for card name
    //Spades
        //2
        //3
        //4
        //5
        //6
        //7
        //8
        //9
        //10
        //Jack
        //Queen
        //King
        //Ace
    //Hearts
        //2
        //3
        //4
        //5
        //6
        //7
        //8
        //9
        //10
        //Jack
        //Queen
        //King
        //Ace
    //Clubs
        //2
        //3
        //4
        //5
        //6
        //7
        //8
        //9
        //10
        //Jack
        //Queen
        //King
        //Ace
    //Diamonds
        //2
        //3
        //4
        //5
        //6
        //7
        //8
        //9
        //10
        //Jack
        //Queen
        //King
        //Ace
    //return card name
//*********************************************//
//*            Hand Ranking                  *//
//*********************************************//
    //declare variables
    //high card hand value= 1
    //temp
    // straight = n, flush = n
    //flush = 6
     // if all spades
     // if all hearts
     // if all clubs
     //if all diamonds
    //get rid of suits
    //for card value over 13. take away 13
    //find high card for flush
    //pair = 2
        //get high card
    //two pair = 3
        //get high card
    //three of a kind = 4
       //get high card
    //sort for straight
        //sort first value
        //sort second value
        //sort third value
        //sort fourth value
    //straight = 5
        //get high card
    //full house = 7
        //get high card
    //four of a kind = 8
        //get high card
    //straight flush = 9
    // if flush and straight then value =9
    //return hand value
//*********************************************//
```

```
//*            Computer Hand                */
//*********************************************//
    //declare string for hand type
    //if 1 then high card
    //if 2 then pair
    //if 3 then two pair
    //if 4 then 3 of a kind
    //if 5 then straight
    //if 6 then flush
    //if 7 then full house
    //if 8 then four of a kind
    //if 9 then straight flush
    //return string for hand type
//*********************************************//
//*              High Card/Sort             */
//*********************************************//
    //temp variables for swap
    //get rid of suits
    //take away 13 for values above 13
    //sort first value
        //swap if needed
    //sort second value
        //swap if needed
    //sort third value
        //swap if needed
    //sort fourth value
        //swap if needed
//*********************************************//
//*              Winning Hand               */
//*********************************************//
    //find top hand value
    //declare variables for ties
    //find all that have the same value of the top hand
    //compare ties for top hand
    //if two have tie
        //win =1
        //win =2
    //if 3 have tie
        //win 1
        //win 2
        //win 3
    //if 4 tie
        //win 1
        //win 2
        //win 3
        //win 4
    //if no tie
        //win 1
        //win 2
        //win 3
        //win 4
    //return win
//*********************************************//
//*           Winning Player            */
//*********************************************//
    //declare string variable for winner
    //if 1 then winner = user
    //if 2 then winner = tom
    //if 3 then winner = howard
```

```
// if 4 then winner = phil
// else tie game
//return winner
```

Flowcharts (More included in the "flowcharts" folder)


Project 1
Straight Poker (main)
Branden Hitt 4/22/15 play 4 player poker
System Libraries iostream ctime cstdlib iomanip string fstream
User Libraries
Global Constants
Function Prototypes rules(), iniDeck(short, short []), round(string, short []), deal1(short,short[]), ranking(short,short,short short,short,short &), airRank(short), cardVal(short), hiCard(short&,short&,short &,short&,short&) winHand(short,short,short, short,short,short, short,short) winner(string,short)
main
Declare ofstream fout;
fout.open("Results. Dat")
srand(static_cast<unsigned int> (time(0)))
Declare info,repeat,p1Rw=0,p2Rw=0,p3Rw=0,p4Rw=0,winner,player1,SIZE =53,rWin=0,perWin,totR
Declare short deck[SIZE]={};
iniDeck(deck,SIZE)
Prompt/Input info
info=='Y'||info=='y'
true
rules();
Input player1
A
rWin=round(player1,deck)
rWin==1
p1Rw++
rWin==2
p2Rw++
rWin==3
p3Rw++
rWin==4
p4Rw++
true
totR++
Input repeat
repeat=='Y'||repeat=='y'
winner=player+" ";
highRW=p1Rw
p2Rw>highRW
winner="Tom Dwan ", highRW=p2Rw
p2Rw==highRW
winner+="& Tom Dwan "
B
p3Rw>highRW
winner="Howard Lederer ", highRW=p3Rw
p3Rw==highRW
winner+="& Howard Lederer "
p4Rw>highRW
winner="Phil Ivey ", highRW=p4Rw
p4Rw==highRW
winner+="& Phil Ivey "
C
Output winner,totR,highRW
perWin= ((p1Rw*1.0f)/totR)* 100.0f
Output to File player1,p1Rw,totR,perWin
fout.close()
end


Project 1
(round function)
Start round
Declare c1=0,c2=0,c3=0,c4=0,c5=0
Declare usrHand,ai1Hand,ai2Hand,ai3Hand
Declare ai1c1=0,ai1c2=0,ai1c3=0,ai1c4=0,ai1c5=0
Declare ai2c1=0,ai2c2=0,ai2c3=0,ai2c4=0,ai2c5=0
Declare ai3c1=0,ai3c2=0,ai3c3=0,ai3c4=0,ai3c5=0
Declare hc=0,ai1hc=0,ai2hc=0,ai3hc=0,winNum
Declare ai1R,ai2R,ai3R,plR
deal1(c1,x) deal1(c2,x) deal1(c3,x) deal1(c4,x) deal1(c4,x)
deal1(ai1c1,x) deal1(ai1c2,x) deal1(ai1c3,x) deal1(ai1c4,x) deal1(ai1c4,x)
deal1(ai2c1,x) deal1(ai2c2,x) deal1(ai2c3,x) deal1(ai2c4,x) deal1(ai2c4,x)
deal1(ai3c1,x) deal1(ai3c2,x) deal1(ai3c3,x) deal1(ai3c4,x) deal1(ai3c4,x)
usrHand=ranking(c1,c2,c3,c4,c5,hc)
ai1Hand=ranking(ai1c1,ai1c2,ai1c3 ,ai1c4,ai1c5,ai1hc)
ai2Hand=ranking(ai2c1,ai2c2,ai2c3 ,ai2c4,ai2c5,ai1hc)
ai3Hand=ranking(ai3c1,ai3c2,ai3c3 ,ai3c4,ai3c5,ai3hc)
plR=aiRank(usrHand)
ai1R=aiRank(ai1Hand)
ai2R=aiRank(ai2Hand)
ai3R=aiRank(ai3Hand)
usrHand==1
Output highcard
usrHand==2
Output pair
usrHand==3
Output two pair
usrHand==4
Output three of a kind
usrHand==5
Output straight
usrHand==6
Output flush
usrHand==7
Output full house
usrHand==8
Output 4 of a kind
usrHand==9
Output straight flush
true
A
cardVal(c1) cardVal(c2) cardVal(c3) cardVal(c4) cardVal(c5)
cardVal(ai1c1) cardVal(ai1c2) cardVal(ai1c3) cardVal(ai1c4) cardVal(ai1c5)
cardVal(ai2c1) cardVal(ai2c2) cardVal(ai2c3) cardVal(ai2c4) cardVal(ai2c5)
cardVal(ai3c1) cardVal(ai3c2) cardVal(ai3c3) cardVal(ai3c4) cardVal(ai3c5)
winNum=winHand(usrHand,ai1Hand,ai2Hand,ai3Hand,hc,ai1hc,ai2hc,ai3hc)
string rndWinr=winner(z,winNum)
Output rndWinr
return winNum
End

Concepts from the Book (Savitch 9th Ed)

Chapter	Concept	Description	Location
1. Introduction	#include <iostream>	System library for input and output	Line 9
	using namespace std;	Used standard naming	Line 15

	Int main( ){
	}
		Main function	Line 33

	char info,repeat;	Variable declaration	Line 38
	cout<<	Output	Line 54
	cin>>	Input	Line 55
	char	Character type variable	Line 38
2. Basics	int	Int type variable	Line 44
	int <u>totR</u>	identifier	Line 44
	x=y*z	Assignment statement	Line 98
	endl;	End line	Line 95
	float	Floating point type variable	Line 44
	short	Short Integer	Line 39
	#include <string>	String library	Line 13
	string player1;	String variable	Line 40
	bool	True or false variable	Line 115
	* / operators	Multiply or divide	Line 98
	+- operators	Add or Subtract	Line 308
	% operator	Mod operator for remainder	Line 308
	if()else()	If-else bool statements	Line 309
	= =	Comparison Operator "equal to"	Line 309
	>,<,>= ,<=	Comparison Operators less than,greater than, less than equal to, greater than equal to	Line 390
	&&	And Operator	Line 390
	||	Or Operator	Line 411
	do{}while();	Do something while condition is satisfied (do-while loop)	Line 307

	i++,i--;	Increment, decrement operators	Line 205
	//something	Comment	Line 113
	const short SIZE=53;	Constant variable	Line 41
	#include <iomanip>	Library for formatting	Line 12
	fixed<<setprecision(2)<< showpoint	Formatting for numbers	Line 100
3. Multi-way Branches	If(if ())	Multi-way if statements	Line 541
	switch(menu){ cases }	Switch statement for menu	Line 135
	for(i;i<n;i++)	For loop	Line 205
	break;	Leaves a loop or switch	Line 139
4. Procedural Abstraction and Functions that return a value	#include <ctime>	ctime library for computer time	Line 10
	srand();	Set the random seed	Line 37
	rand();	Get random number	Line 308
	static_cast<unsigned int>	Type cast	Line 37
	#include <cstdlib>	Library used for random seed	Line 11
	short round();	Function that returns a value	Line 24
	short hiCard(short &);	Function with a pass-by-reference	Line 29
5. Functions for all Subtasks	void rules();	Void function	Line 22
	return;	Ends function	Line 107
	string aiRank(short);	Call by value in function	Line 27
6. Streams and Basic I/O	ofstream fout;	Declares out file stream variable	Line 35
	#include <fstream>	fstream library for file usage	Line 14
	fout.open	Open file	Line 36
	fout.close	Close file	Line 105
7. Intro to Arrays	short deck[Size]={};	Array declaration	Line 42
	void deal1(short, short [])	Array in a function	Line 25

References

1.Textbook - Problem Solving with C++, Savitch 9th Ed
2. PokerStars.com – (For card hand rankings list)

Program Code

```
/*
 * File:   main.cpp
 * Author: Branden Hitt
 * Created on May 6, 2015, 1:02 PM
 *      Purpose: Code a game of straight poker
 */

//System Libraries
#include <iostream>//I/O standard
#include <ctime>//computer time
#include <cstdlib>//for random seed
#include <iomanip>//formatting
#include <string>//for strings
#include <fstream>//file in/out
using namespace std;

//User Libraries

//Global Constants

//Function Prototypes
void rules();//instructions on how to play
void iniDeck(short [],short);//initializes and shuffles the deck
short round(string ,short []);//starts a new round of poker
void deal1(short &,short []);//deal 1 card
short ranking(short ,short ,short ,short ,short,short &);//gives hand value
string aiRank(short); //finds ranking of computer hand
string cardVal(short);//finds what the card is for output
void hiCard(short &,short &,short &,short &,short &);//sort for high card
short winHand(short,short,short,short,short,short,short,short);//find winning hand
string winner(string,short);//find round winner
//Execution Begins Here!
int main(int argc, char** argv) {
   //Declare Variables
   ofstream fout;//output to file
   fout.open("Results.Dat");//open file
   srand(static_cast<unsigned int>(time(0)));//set random seed
   char info,repeat;//input for info query, and repeat query
   short p1Rw=0,p2Rw=0,p3Rw=0,p4Rw=0;//total wins for each player
   string winner, player1;//overall winner and player name input
   const short SIZE=53;//size of the deck
   short deck[SIZE]={};//declare array with zeros
   short rWin=0;//current round win
   float perWin; int totR=0;//percentage of wins for the player/ totals rounds
   iniDeck(deck,SIZE);//fill array with card values (1-52)
   cout<<"This is a table for Straight Poker."<<endl;
   //Prompt for rules
   cout<<"If you would like instructions on how to play, then enter in 'I' now:"<<endl;
   cout<<"otherwise, enter in 'N'"<<endl;
   cin>>info;
   if(info=='I'||info=='i') rules();
   //prompt for name of player
   cout<<"************************"<<endl;
   cout<<"Please first enter in your name:"<<endl;
```

```
    cin>>player1;
    cout<<"*************************"<<endl;
    cout<<"Hello, "<<player1<<"! Welcome to the table."<<endl;
    cout<<"Please hit enter to begin"<<endl;
    cin.get();
    cin.ignore();
    cout<<"*************************"<<endl;
    cout<<"Now that you are seated, why don't you meet the other players."<<endl;
    cout<<"To your left is Tom Dwan, across from you is Howard Lederer, and to"<<endl;
    cout<<"your right is Phil Ivey. Good luck players."<<endl;
    cout<<"Please hit enter twice to continue"<<endl;
    cin.get();
    cin.ignore();
    do{//do
        //Start new round
        rWin=round(player1,deck);
        if(rWin==1) p1Rw++;//if 1 player 1 win round
        if(rWin==2) p2Rw++;//if 2 player 2 win round
        if(rWin==3) p3Rw++;//if player 3 win round
        if(rWin==4) p4Rw++;//if player 4 win round
        //Shuffle
        iniDeck(deck,SIZE);
        totR++;//increment the total rounds
        cout<<"*************************"<<endl;
        cout<<"Would you like to play another hand?"<<endl;
        cout<<"Enter in Y for yes or N for no:"<<endl;
        cin>>repeat;//prompt for new round
    }while(repeat=='Y'||repeat=='y');//while player wants to repeat
    //Congratulate Overall Winner
    cout<<"*************************"<<endl;
    winner=player1+" ";
    short highRW=p1Rw;
    if(p2Rw>highRW) winner="Tom Dwan ", highRW=p2Rw;
    else if(p2Rw==highRW) winner+="& Tom Dwan ";
    if(p3Rw>highRW) winner="Howard Lederer ", highRW=p3Rw;
    else if(p3Rw==highRW) winner+="& Howard Lederer ";
    if(p4Rw>highRW) winner="Phil Ivey ", highRW=p4Rw;
    else if(p4Rw==highRW) winner+="& Phil Ivey ";
    cout<<"Congrats to "<<winner<<"for being the table's big winner(s)."<<endl;
    cout<<"Out of "<<totR<<" total round(s), "<<winner<<"won "<<highRW<<" round(s) of Poker."<<endl;
    cout<<"Thank you for playing!"<<endl;
    cout<<endl;
    cout<<"(Your results can be found in Results.dat)"<<endl;
    perWin=((p1Rw*1.0f)/totR)*100.0f;
    //output results to a file
    fout<<fixed<<setprecision(2)<<showpoint;
    fout<<"Player name : "<<player1<<endl;
    fout<<"Round Wins  : "<<p1Rw<<endl;
    fout<<"Total Rounds: "<<totR<<endl;
    fout<<"Percent Win : %"<<perWin;
    fout.close();
    //Exit Stage Right!
    return 0;
}
//*******************************************//
//*            Rules Function              *//
//*******************************************//
void rules(){
    //declare variables
```

```
char menu,cho='N';//menu / choice
bool skip=false;//bool for skip
cout<<"*************************"<<endl;
cout<<"How to Play:"<<endl;//begin explanation of how to play
cout<<"Each player is dealt 5 random cards (or a Hand) from a standard 52-card deck."<<endl;
cout<<"Whoever has the best combination of cards based on rankings will win the round."<<endl;
cout<<"The more rare the hand, the higher the ranking will be."<<endl;
cout<<"The worst hand is nothing special with just a High Card, while the best is a Royal Flush."<<endl;
cout<<"In the event that two players have the same hand, the higher card (pair,flush,etc)"<<endl;
cout<<"will win."<<endl;
 do{//do
cout<<"The Hand Rankings are as followed:"<<endl;//start showing hand rankings
cout<<"0.High Card | 1.Pair | 2.Two Pair | 3.Three of a Kind | 4.Straight "<<endl;
cout<<"5.Flush | 6.Full House | 7.Four of a Kind | 8.Straight Flush | 9.Royal Flush"<<endl;
cout<<"*************************"<<endl;
cout<<"To view an example/explanation of a specific hand, enter in"<<endl;
cout<<"the corresponding number. (for example: enter in 4 to view a straight)"<<endl;
cout<<"Otherwise, enter in S to get started:"<<endl;
cout<<"*************************"<<endl;
cin>>menu;//prompt for specific hand ranking display
cout<<"*************************"<<endl;
switch(menu){//menu for hand display
   case'0':{//case for hi card
      cout<<"High Card: This hand has nothing to rely on but a High card."<<endl;
      cout<<"Ex: 3 Diamonds | Jack Clubs | 8 Spades | 4 Hearts | 2 Spades"<<endl;
      break;
   }
   case'1':{//case for pair
      cout<<"Pair: Two cards of the same rank."<<endl;
      cout<<"Ex: Ace Hearts | Ace Diamonds | 8 Clubs | 4 Spades | 7 Hearts"<<endl;
      break;
   }
   case'2':{//case for two pair
      cout<<"Two Pair: Two different pairs of cards."<<endl;
      cout<<"Ex: 4 Spades | 4 Clubs | 3 Clubs | 3 Diamonds | Queen Clubs"<<endl;
      break;
   }
   case'3':{//case for three of a kind
      cout<<"Three of a Kind: Three cards of the same rank."<<endl;
      cout<<"Ex: 7 Clubs | 7 Diamonds | 7 Spades | King Clubs | 3 Diamonds"<<endl;
      break;
   }
   case'4':{//case for straight
      cout<<"Straight: Five cards in a sequence, but not of the same suit."<<endl;
      cout<<"Ex: 5 Hearts | 6 Diamonds | 7 Spades | 8 Diamonds | 9 Clubs"<<endl;
      break;
   }
   case'5':{//case for flush
      cout<<"Flush: Any five cards of the same suit, but not in a sequence."<<endl;
      cout<<"Ex: 4 Spades | Jack Spades | 8 Spades | 2 Spades | 9 Spades"<<endl;
      break;
   }
   case'6':{//case for full house
      cout<<"Full House: Three of a Kind and a Pair."<<endl;
      cout<<"Ex: 10 Hearts | 10 Diamonds | 10 Spades | 9 Clubs | 9 Diamonds"<<endl;
      break;
   }
   case'7':{//case for four of a kind
      cout<<"Four of a Kind: All four cards of the same rank."<<endl;
```

```
        cout<<"Ex: Jack Hearts | Jack Spades | Jack Diamonds | Jack Clubs | 7 Hearts"<<endl;
        break;
      }
      case'8':{//case for straight flush
        cout<<"Straight Flush: Five cards in a sequence, all in the same suit."<<endl;
        cout<<"Ex: 4 Spades | 5 Spades | 6 Spades | 7 Spades | 8 Spades"<<endl;
        break;
      }
      case'9':{//case for royal flush
        cout<<"Royal Flush: 10, J, Q, K, A (all in the same suit)."<<endl;
        cout<<"Ex: 10 Spades | Jack Spades | Queen Spades | King Spades | Ace Spades"<<endl;
        break;
      }
      default:{//default for skip
        cho=='N';
        skip=true;//set skip to true
        break;
      }
    }
    if(skip==false){//if skip is false
    cout<<"*************************"<<endl;
    cout<<"Would you like to view another Hand Ranking?"<<endl;
    cout<<"Enter in Y for yes or N for no:"<<endl;
    cin>>cho;//prompt for repeat
    cout<<"*************************"<<endl;
    }
    }while(cho=='Y'||cho=='y');//while player wants to repeat
}
//*********************************************//
//*          Initialize Deck            *//
//*********************************************//
void iniDeck(short cards[],short n){
    for(int i=0;i<n;i++){//increment while still in array
        cards[i]=i;//for position i, set i to i ex: 41 =41
    }
}
//*********************************************//
//*            New round                *//
//*********************************************//
short round(string z,short x[]){
    //Declare variables
    short c1=0,c2=0,c3=0,c4=0,c5=0;//player cards
    short usrHand,ai1Hand,ai2Hand,ai3Hand;//hand values
    short ai1c1=0,ai1c2=0,ai1c3=0,ai1c4=0,ai1c5=0;//computer 1 hand
    short ai2c1=0,ai2c2=0,ai2c3=0,ai2c4=0,ai2c5=0;//computer 2 hand
    short ai3c1=0,ai3c2=0,ai3c3=0,ai3c4=0,ai3c5=0;//computer 3 hand
    short hc=0,ai1hc=0,ai2hc=0,ai3hc=0;//high card for tie hands
    short winNum;//player number who has top hand
    string ai1R,ai2R,ai3R,plR; //quick string for hand ranking
    cout<<"*************************"<<endl;
    //Deal individual hands
    deal1(c1,x);
    deal1(c2,x);
    deal1(c3,x);
    deal1(c4,x);
    deal1(c5,x);
    deal1(ai1c1,x);
    deal1(ai1c2,x);
    deal1(ai1c3,x);
```

```
deal1(ai1c4,x);
deal1(ai1c5,x);
deal1(ai2c1,x);
deal1(ai2c2,x);
deal1(ai2c3,x);
deal1(ai2c4,x);
deal1(ai2c5,x);
deal1(ai3c1,x);
deal1(ai3c2,x);
deal1(ai3c3,x);
deal1(ai3c4,x);
deal1(ai3c5,x);
//high card and sort
hiCard(c1,c2,c3,c4,c5);
hiCard(ai1c1,ai1c2,ai1c3,ai1c4,ai1c5);
hiCard(ai2c1,ai2c2,ai2c3,ai2c4,ai2c5);
hiCard(ai3c1,ai3c2,ai3c3,ai3c4,ai3c5);
//find hand value
usrHand=ranking(c1,c2,c3,c4,c5,hc);
ai1Hand=ranking(ai1c1,ai1c2,ai1c3,ai1c4,ai1c5,ai1hc);
ai2Hand=ranking(ai2c1,ai2c2,ai2c3,ai2c4,ai2c5,ai2hc);
ai3Hand=ranking(ai3c1,ai3c2,ai3c3,ai3c4,ai3c5,ai3hc);
//set string for ai hand ranking
plR=aiRank(usrHand);
ai1R=aiRank(ai1Hand);
ai2R=aiRank(ai2Hand);
ai3R=aiRank(ai3Hand);
//state player hand value
if(usrHand==1)cout<<"Your hand value is low with just a high card ("<<cardVal(c5)<<")."<<endl;
if(usrHand==2)cout<<"Your hand value is average with just a pair."<<endl;
if(usrHand==3)cout<<"You have two pair."<<endl;
if(usrHand==4)cout<<"You have three of a kind. Not too bad."<<endl;
if(usrHand==5)cout<<"Congrats! You have a straight."<<endl;
if(usrHand==6)cout<<"Wow! You have a Flush."<<endl;
if(usrHand==7)cout<<"What a strong hand. You have a Full House!"<<endl;
if(usrHand==8)cout<<"Four of a Kind! Quick, don't let them see your excitement."<<endl;
if(usrHand==9)cout<<"A Straight Flush! You can't be beat!"<<endl;
//Display player hand
cout<<"Your Hand: "<<cardVal(c1)<<"|"<<cardVal(c2)<<"|"<<cardVal(c3)<<
        "|"<<cardVal(c4)<<"|"<<cardVal(c5)<<endl;
cout<<"Please hit enter twice to see everyone's hands:"<<endl;
cin.get();
cin.ignore();
//Unveil all hands
cout<<"*************************"<<endl;
cout<<"              All Player Hands     "<<endl;
cout<<"Your Hand: "<<cardVal(c1)<<"|"<<cardVal(c2)<<"|"<<cardVal(c3)<<
        "|"<<cardVal(c4)<<"|"<<cardVal(c5)<<"  ("<<plR<<")"<<endl;
cout<<"Tom's Hand: "<<cardVal(ai1c1)<<"|"<<cardVal(ai1c2)<<"|"<<cardVal(ai1c3)<<
        "|"<<cardVal(ai1c4)<<"|"<<cardVal(ai1c5)<<"  ("<<ai1R<<")"<<endl;
cout<<"Howard's Hand: "<<cardVal(ai2c1)<<"|"<<cardVal(ai2c2)<<"|"<<cardVal(ai2c3)<<
        "|"<<cardVal(ai2c4)<<"|"<<cardVal(ai2c5)<<"  ("<<ai2R<<")"<<endl;
cout<<"Phil's Hand: "<<cardVal(ai3c1)<<"|"<<cardVal(ai3c2)<<"|"<<cardVal(ai3c3)<<
        "|"<<cardVal(ai3c4)<<"|"<<cardVal(ai3c5)<<"  ("<<ai3R<<")"<<endl;
cout<<"Please hit enter twice to find the winning hand:"<<endl;
cin.get();
cin.ignore();
cout<<"*************************"<<endl;
//find winning hand
```

```
    winNum=winHand(usrHand,ai1Hand,ai2Hand,ai3Hand,hc,ai1hc,ai2hc,ai3hc);//find winNum
    string rndWinr=winner(z,winNum);//find winner name
    cout<<rndWinr<<" has the winning hand. "<<endl;//state winner
    cout<<"Please hit enter twice to continue:"<<endl;
    cin.get();
    cin.ignore();
    cout<<"************************"<<endl;
    return winNum;//return winner number
}
//********************************************//
//*              Deal 1                    *//
//********************************************//
void deal1(short &c1,short y[]){
    //declare variables
    char repeat;//repeat
    short card;//new card
    do{//do
        card=rand()%52+1;//pick random number
        if(y[card]==0){//if card is 0, then its already picked, so repeat
            repeat='Y';
        }else{//else dont repeat
            repeat='N';
        }
    }while(repeat=='Y');//while need new card
    y[card]=0;//set the card you picked to zero in array
    c1=card;//c1 = card
}
//********************************************//
//*            Card Values                 *//
//********************************************//
string cardVal(short v){
    string f;//declare string f for card name
    //Spades
    if(v==1)f="2 Spades";
    if(v==2) f="3 Spades";
    if(v==3) f="4 Spades";
    if(v==4) f="5 Spades";
    if(v==5) f="6 Spades";
    if(v==6) f="7 Spades";
    if(v==7) f="8 Spades";
    if(v==8) f="9 Spades";
    if(v==9) f="10 Spades";
    if(v==10) f="Jack Spades";
    if(v==11) f="Queen Spades";
    if(v==12) f="King Spades";
    if(v==13) f="Ace Spades";
    //Hearts
    if(v==14) f="2 Hearts";
    if(v==15) f="3 Hearts";
    if(v==16) f="4 Hearts";
    if(v==17) f="5 Hearts";
    if(v==18) f="6 Hearts";
    if(v==19) f="7 Hearts";
    if(v==20) f="8 Hearts";
    if(v==21) f="9 Hearts";
    if(v==22) f="10 Hearts";
    if(v==23) f="Jack Hearts";
    if(v==24) f="Queen Hearts";
    if(v==25) f="King Hearts";
    if(v==26) f="Ace Hearts";
    //Clubs
    if(v==27) f="2 Clubs";
    if(v==28) f="3 Clubs";
    if(v==29) f="4 Clubs";
    if(v==30) f="5 Clubs";
    if(v==31) f="6 Clubs";
    if(v==32) f="7 Clubs";
    if(v==33) f="8 Clubs";
    if(v==34) f="9 Clubs";
    if(v==35) f="10 Clubs";
    if(v==36) f="Jack Clubs";
    if(v==37) f="Queen Clubs";
    if(v==38) f="King Clubs";
    if(v==39) f="Ace Clubs";
    //Diamonds
    if(v==40) f="2 Diamonds";
    if(v==41) f="3 Diamonds";
    if(v==42) f="4 Diamonds";
    if(v==43) f="5 Diamonds";
    if(v==44) f="6 Diamonds";
    if(v==45) f="7 Diamonds";
    if(v==46) f="8 Diamonds";
    if(v==47) f="9 Diamonds";
    if(v==48) f="10 Diamonds";
    if(v==49) f="Jack Diamonds";
    if(v==50) f="Queen Diamonds";
    if(v==51) f="King Diamonds";
    if(v==52) f="Ace Diamonds";
    return f;//return card name
}
//*********************************************//
//*            Hand Ranking                 *//
//*********************************************//
short ranking(short a,short b,short c,short d,short e, short &hi1){
    //declare variables
    //high card hand value= 1
    short value=1,temp;//temp
    char str='N',flu='N';// straight = n, flush = n
    //flush = 6
    if((a>=1&&a<=13)&&(b>=1&&b<=13)&&(c>=1&&c<=13)&&(d>=1&&d<=13)&&(e>=1&&e<=13)) value=6,flu='Y'; //spades
    if((a>=14&&a<=26)&&(b>=14&&b<=26)&&(c>=14&&c<=26)&&(d>=14&&d<=26)&&(e>=14&&e<=26)) value=6,flu='Y';
//hearts
    if((a>=27&&a<=39)&&(b>=27&&b<=39)&&(c>=27&&c<=39)&&(d>=27&&d<=39)&&(e>=27&&e<=39)) value=6,flu='Y';
//clubs
    if((a>=40&&a<=52)&&(b>=40&&b<=52)&&(c>=40&&c<=52)&&(d>=40&&d<=52)&&(e>=40&&e<=52)) value=6,flu='Y';
//diamonds
    //get rid of suits
    for(a;a>13;a-=13);
    for(b;b>13;b-=13);
    for(c;c>13;c-=13);
    for(d;d>13;d-=13);
    for(e;e>13;e-=13);
    //high card for flush
    if(value==6) hi1=e;
    //pair = 2
    if(a==b||a==c||a==d) value=2,hi1=a;
    if(a==e)value=2,hi1=a;
    if(b==c||b==d) value=2,hi1=b;
    if(c==d) value=2,hi1=c;
    if(c==e) value=2,hi1=c;
    if(d==e) value=2,hi1=e;
    if(b==e) value=2,hi1=e;
    //two pair = 3
    if((b==c&&d==e)||(b==d&&c==e)||(b==e&&c==d)) value=3,hi1=b;
    if((a==c&&d==e)||(a==e&&c==d)||(a==d&&c==e)) value=3,hi1=a;
    if((a==b&&d==e)||(a==e&&b==d)||(a==d&&b==e)) value=3,hi1=a;
    if((a==b&&c==e)||(a==e&&b==c)||(a==c&&b==e)) value=3,hi1=a;
    if((a==b&&c==d)||(a==c&&b==d)||(a==d&&b==c)) value=3,hi1=a;
    //three of a kind = 4
    if((a==d&&a==e)||(a==b&&a==e)||(a==b&&a==c)) value=4,hi1=a;
    if((c==d&&c==e)||(b==d&&b==e)) value=4,hi1=e;
    if((a==c&&a==e)||(a==b&&a==d)) value=4,hi1=a;
    if((b==c&&b==e)||(a==c&&a==d)) value=4,hi1=c;
    if(b==c&&b==d) value=4,hi1=b;
    //sort for straight
        //sort first value
        if(a>b) temp=a,a=b,b=temp;
        if(a>c) temp=a,a=c,c=temp;
        if(a>d) temp=a,a=d,d=temp;
        if(a>e) temp=a,a=e,e=temp;
        //sort second value
        if(b>c) temp=b,b=c,c=temp;
        if(b>d) temp=b,b=d,d=temp;
        if(b>e) temp=b,b=e,e=temp;
        //sort third value
        if(c>d) temp=c,c=d,d=temp;
        if(c>e) temp=c,c=e,e=temp;
        //sort fourth value
        if(d>e) temp=d,d=e,e=temp;
    //straight = 5
    if((e-d==1)&&(d-c==1)&&(c-b==1)&&(b-a==1)) value=5,str='Y',hi1=e;
    if((a==1)&&(b==2)&&(c==3)&&(d==4)&&(e==13)) value=5,str='Y',hi1=e;
    //full house = 7
    if((c==d&&c==e&&a==b)||(a==d&&a==e&&b==c)) value=7,hi1=d;
    if((a==b&&a==e&&c==d)||(a==b&&a==c&&d==e)) value=7,hi1=a;
    if((a==c&&a==e&&b==d)||(a==b&&a==d&&c==e)) value=7,hi1=a;
    if((b==d&&b==e&&a==c)||(b==c&&b==d&&a==e)) value=7,hi1=b;
    if((b==c&&b==e&&a==d)||(a==c&&a==d&&b==e)) value=7,hi1=c;
    //four of a kind = 8
    if((b==c&&d==e&&c==d)||(a==c&&d==e&&c==d)||(a==b&&d==e&&b==d)) value=8,hi1=d;
    if((a==b&&c==e&&b==c)||(a==b&&c==d&&b==c)) value=8,hi1=a;
    //straight flush = 9
    if(flu=='Y'&&str=='Y') value=9;// if flush and straight then value =9
    return value;//return hand value
}
//*******************************************//
//*            Computer Hand               *//
//*******************************************//
string aiRank(short a){
    string state;//declare string for hand type
    if(a==1)state="High card";//if 1 then high card
    if(a==2)state="Pair";//if 2 then pair
    if(a==3)state="Two Pair";//if 3 then two pair
    if(a==4)state="Three of a Kind";// if 4 then 3 of a kind
    if(a==5)state="Straight";//if 5 then straight
    if(a==6)state="Flush";//if 6 then flush
    if(a==7)state="Full House";// 7 then full house
    if(a==8)state="Four of a Kind";//if 8 then four of a kind
    if(a==9)state="Straight Flush";// if 9 then straight flush
    return state;//return string for hand type
}
//*********************************************//
//*              High Card/Sort              *//
//*********************************************//
void hiCard(short &a,short &b,short &c,short &d,short &e){
    short tempa=a,tempb=b,tempc=c,tempd=d,tempe=e;//temp variables for swap
    short temp=0;
    //get rid of suits
    for(tempa;tempa>13;tempa-=13);//take away 13 for values above 13
    for(tempb;tempb>13;tempb-=13);
    for(tempc;tempc>13;tempc-=13);
    for(tempd;tempd>13;tempd-=13);
    for(tempe;tempe>13;tempe-=13);
    //sort first value
    if(tempa>tempb){
        temp=tempa,tempa=tempb,tempb=temp;
        temp=a,a=b,b=temp;//swap if needed
    }
    if(tempa>tempc){
        temp=tempa,tempa=tempc,tempc=temp;
        temp=a,a=c,c=temp;
    }
    if(tempa>tempd){
        temp=tempa,tempa=tempd,tempd=temp;
        temp=a,a=d,d=temp;
    }
    if(tempa>tempe){
        temp=tempa,tempa=tempe,tempe=temp;
        temp=a,a=e,e=temp;
    }
    //sort second value
    if(tempb>tempc){//swap if needed
        temp=tempb,tempb=tempc,tempc=temp;
        temp=b,b=c,c=temp;
    }
    if(tempb>tempd){
        temp=tempb,tempb=tempd,tempd=temp;
        temp=b,b=d,d=temp;
    }
    if(tempb>tempe){
        temp=tempb,tempb=tempe,tempe=temp;
        temp=b,b=e,e=temp;
    }
    //sort third value
    if(tempc>tempd){
        temp=tempc,tempc=tempd,tempd=temp;
        temp=c,c=d,d=temp;//swap if needed
    }
    if(tempc>tempe){
        temp=tempc,tempc=tempe,tempe=temp;
        temp=c,c=e,e=temp;
    }
    //sort fourth value
    if(tempd>tempe){
        temp=tempd,tempd=tempe,tempe=temp;
        temp=d,d=e,e=temp;//swap if needed
```

```
  }
}
//*********************************************//
//*              Winning Hand                *//
//*********************************************//
short winHand(short a,short b,short c,short d,short h1,short h2,short h3,short h4){
  //find top hand value
  short topHand=a,pl1=0,pl2=0,pl3=0,pl4=0,win;//declare variables for ties
  if(b>topHand) topHand=b;
  if(c>topHand) topHand=c;
  if(d>topHand) topHand=d;
  //find all that have the same value of the top hand
  if(a==topHand) pl1=1;
  if(b==topHand) pl2=1;
  if(c==topHand) pl3=1;
  if(d==topHand) pl4=1;
  //compare ties for top hand
  if((pl1==1)&&(pl2==1)){//if two have tie
    if(h1>h2) win=1;//win =1
    if(h2>h1) win=2;//win =2
  }
  if((pl1==1)&&(pl3==1)){
    if(h1>h3) win=1;
    if(h3>h1) win=3;
  }
  if((pl1==1)&&(pl4==1)){
    if(h1>h4) win=1;
    if(h4>h1) win=4;
  }
  if((pl2==1)&&(pl3==1)){
    if(h2>h3) win=2;
    if(h3>h2) win=3;
  }
  if((pl2==1)&&(pl4==1)){
    if(h2>h4) win=2;
    if(h4>h2) win=4;
  }
  if((pl3==1)&&(pl4==1)){
    if(h3>h4) win=3;
    if(h4>h3) win=4;
  }
  if((pl1==1)&&(pl2==1)&&(pl3==1)){//if 3 have tie
    if(h1>h2&&h1>h3) win=1;//win 1
    if(h2>h1&&h2>h3) win=2;//win 2
    if(h3>h1&&h3>h2) win=3;//win 3
  }
  if((pl1==1)&&(pl3==1)&&(pl4==1)){
    if(h1>h4&&h1>h3) win=1;
    if(h4>h1&&h4>h3) win=4;
    if(h3>h1&&h3>h4) win=3;
  }
  if((pl1==1)&&(pl2==1)&&(pl4==1)){
    if(h1>h2&&h1>h4) win=1;
    if(h2>h1&&h2>h4) win=2;
    if(h4>h1&&h4>h2) win=4;
  }
  if((pl3==1)&&(pl2==1)&&(pl4==1)){
    if(h3>h2&&h3>h4) win=3;
    if(h2>h3&&h2>h4) win=2;
```

```
        if(h4>h3&&h4>h2) win=4;
    }
    if((pl1==1)&&(pl2==1)&&(pl3==1)&&(pl4==1)){//if 4 tie
        if(h1>h2&&h1>h3&&h1>h4) win=1;//win 1
        if(h2>h1&&h2>h3&&h2>h4) win=2;//win 2
        if(h3>h2&&h3>h1&&h3>h4) win=3;//win 3
        if(h4>h2&&h4>h3&&h4>h1) win=4;//win 4
    }
    //if no tie
    if((pl1==1)&&(pl2==0)&&(pl3==0)&&(pl4==0)) win=1;//win 1
    if((pl1==0)&&(pl2==1)&&(pl3==0)&&(pl4==0)) win=2;//win 2
    if((pl1==0)&&(pl2==0)&&(pl3==1)&&(pl4==0)) win=3;//win 3
    if((pl1==0)&&(pl2==0)&&(pl3==0)&&(pl4==1)) win=4;//win 4
    //return win
    return win;
}
//*********************************************//
//*           Winning Player                *//
//*********************************************//
string winner(string you,short a){
    string winner;//declare string variable for winner
    if(a==1) winner=you;//if 1 then winner = user
    if(a==2) winner="Tom";//if 2 then winner = tom
    if(a==3) winner="Howard";//if 3 then winner = howard
    if(a==4) winner="Phil";// if 4 then winner = phil
    if(a<1||a>4) winner="(Tie Game) No one";// else tie game
    return winner;//return winner
}
```