Texas Hold’em

C++ Project 2

Kyle Powell

Cis/csc 5

Summer 2017

45549

Table of Contents

**Rules of the Game 2**

**Flowchart 3**

**Pseudocode 8**

**Check List 10**

**Program 13**

# Rules of the Game

1. A single 52-card deck is used. All cards count as its poker value. Aces may be high or low.
2. Two cards shall be dealt down to each player.
3. The player to the left of the big blind must either call or raise the big blind bet.
4. The small blind may also raise the big blind.
5. Three community cards will be dealt face up in the center of the table. This is called the "flop."
6. Another round of betting will ensue, starting with the player to the dealer's left.
7. A fourth community card will be dealt face up in the center of the table. This card is called the "turn."
8. Another round of betting will ensue, starting with the player to the dealer's left.
9. A fifth and final community card will be dealt face up in the center of the table. This card is called the "river."
10. Another round of betting will ensue, starting with the player to the dealer's left.
11. Each player still in the game at the end will determine the highest poker value among his own two cards and the five community cards. It is NOT a requirement that the player use both of his own cards. The player with the hand of highest poker value shall win. Following are the hand rankings.
    1. Straight flush: Five consecutive and suited cards.
    2. Four of a kind: Four cards of the same rank, plus any fifth card.
    3. Full house: Three of a kind and a pair.
    4. Flush: Any five cards of the same suit, except for a higher ranking straight flush.
    5. Straight: Five consecutive cards, except for a higher ranking straight flush.
    6. Three of a kind: Three cards of the same rank, plus any other two cards.
    7. Two pair: Two pairs, plus any fifth card.
    8. Pair: A pair and any other three cards.
    9. High: Any five cards that do not form any higher poker hand.
12. If two or more players have poker values of the same rank then the individual cards will be used to break the tie. If necessary all five cards will be considered.

# Flowchart

# Pseudocode

File: main.cpp

Author: Kyle Powell

Created on July 24, 2017, 12:45 PM

Purpose: Texas Hold'em

System Libraries

Input - Output Library

Random Number Library

Time Library

String Library

Format Library

File I/O Library

Name-space under which system libraries exist

//User Libraries

//Global Constants

//Function Prototypes

Execution begins here

Texas Hold'em Rules

Note to Player

Set the random number seed

Declare variables

deck of cards

Menu choice

bet related variables

//Loop

Output money

Output cards; for test

Shuffle Cards

Output cards again; for test

Pre-Flop (Deal cards to players)

Output money

The Flop

Output Cards

Output money

The Turn

Output Cards

Output money

The River

Output Cards

Output money

Output all cards

Variables for hand determination (player)

high card

pair, 2 cards with same face value

two pair, 2 different cards with pairs

three of a kind, 3 cards with same face value

flush, 5 cards with the same suit

full house, three of a kind + a pair

four of a kind, 4 cards with same face value

Variables for hand determination (opponent)

high card

pair, 2 cards with same face value

two pair, 2 different cards with pairs

three of a kind, 3 cards with same face value

flush, 5 cards with the same suit

full house, three of a kind + a pair

four of a kind, 4 cards with same face value

Finding pairs in the opponent's hand

counter array

finds pair in opponent's hole cards

finds pairs with opp hole cards in com

finds pairs with opp hole cards in com

finds pairs in com cards

Finding cards with same suit

finds same suit in opponent's hole cards

finds same suit with opponent's hole cards in com

finds same suit with opponent's hole cards in com

finds same suit in com cards

Determine hand

pairs

flush

Finding pairs in the player's hand

Finding cards with same suit in player's hand

finds same suit in player's hole cards

finds same suit with player's hole cards in com

finds same suit with player's hole cards in com

finds same suit in com cards

Determine hand

pairs

flush

Output hands of player and opponent

Compare hands

Exit stage right!

FUNCTIONS

Function for dealing cards and bets

call or fold, bet choice

Deal community cards

Output all visible cards, community cards and player hole cards

Opponent's bet

Input data for bet

pre-flop function

call or fold, bet choice

Opponent's Hole Cards

Opponent's bet

Output Player's hole cards

Input data for bet

Shows the face value of cards

Shows the suit of cards

S -> Spades

D -> Diamond

C -> Clubs

H -> Hearts;

# Check List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Cross Reference for Project 2** | | | |
|  |  |  |  |  |
|  |  |  |  | **Where in Code** |
|  | **Chapter** | **Section** | **Topic** | **Line number** |
|  | 2 | 2 | cout | 65 |
|  |  | 3 | libraries | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
|  |  | 4 | variables/literals | 53 |
|  |  | 5 | Identifiers | 157 |
|  |  | 6 | Integers | 59 |
|  |  | 7 | Characters | 61 |
|  |  | 8 | Strings | 130 |
|  |  | 9 | Floats No Doubles | 62 |
|  |  | 10 | Bools | 60 |
|  |  | 11 | Sizeof \*\*\*\*\* |  |
|  |  | 12 | Variables 7 characters or less | 53-62 |
|  |  | 13 | Scope \*\*\*\*\* No Global Variables |  |
|  |  | 14 | Arithmetic operators | 201 |
|  |  | 15 | Comments 20%+ |  |
|  |  | 16 | Named Constants | 53 |
|  |  | 17 | Programming Style \*\*\*\*\* Emulate |  |
|  |  |  |  |  |
|  | 3 | 1 | cin | 139 |
|  |  | 2 | Math Expression | 185 |
|  |  | 3 | Mixing data types \*\*\*\* |  |
|  |  | 4 | Overflow/Underflow \*\*\*\* |  |
|  |  | 5 | Type Casting | 209 |
|  |  | 6 | Multiple assignment \*\*\*\*\* |  |
|  |  | 7 | Formatting output | 94 |
|  |  | 8 | Strings | 130 |
|  |  | 9 | Math Library | did not use math library |
|  |  | 10 | Hand tracing \*\*\*\*\*\* |  |
|  |  |  |  |  |
|  | 4 | 1 | Relational Operators | 68 |
|  |  | 2 | if | 68 |
|  |  | 4 | If-else | 140-2 |
|  |  | 5 | Nesting | 115 |
|  |  | 6 | If-else-if | 216-23 |
|  |  | 7 | Flags \*\*\*\*\* |  |
|  |  | 8 | Logical operators | 83 |
|  |  | 11 | Validating user input | 418? |
|  |  | 13 | Conditional Operator | 176 |
|  |  | 14 | Switch | 74 |
|  |  |  |  |  |
|  | 5 | 1 | Increment/Decrement | 77 |
|  |  | 2 | While | 78-81 |
|  |  | 5 | Do-while | 93-6 |
|  |  | 6 | For loop | 81 |
|  |  | 11 | Files input/output both | 37/did not use file output |
|  |  | 12 | No breaks in loops \*\*\*\*\*\* |  |
|  |  |  |  |  |
|  | 6 | 3 | Function Prototypes | 319-46 |
|  |  | 5 | Passing by value | 312 |
|  |  | 8 | Returning values from functions | 401-6 |
|  |  | 9 | Returning a boolean \*\*\*\*\*\* |  |
|  |  | 10 | No Global Variables Allowed |  |
|  |  |  | Only Global Constants |  |
|  |  |  | Meaning Conversions,Physical Constants only |  |
|  |  | 11 | Static Local | 320 |
|  |  | 12 | Default arguments | 397 |
|  |  | 13 | Reference Parameters | 348 |
|  |  | 14 | Overloading functions |  |
|  |  | 15 | Exit function \*\*\*\*\*\*\* |  |
|  |  |  |  |  |
|  | 7 | 4 | Array Initialization | 47 |
|  |  | 6 | Processing Arrays | 274 |
|  |  | 7 | Parallel Arrays | 47-55 |
|  |  | 8 | Arrays as function arguments | 348 |
|  |  | 9 | 2-D Arrays | 158 |
|  |  | 12 | STL Vector |  |
|  |  |  |  |  |
|  | 8 | 1 | Linear and Binary Search |  |
|  |  | 3 | Bubble and Selection Sort |  |
|  |  | 5 | Search/Sorting Vectors \*\*\*\*\*\* |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | \*\*\*\*\*\* Not required to show |  |  |  |

# Program

/\*

\* File: main.cpp

\* Author: Kyle Powell

\* Created on July 24, 2017, 12:45 PM

\* Purpose: Texas Hold'em

\*/

//System Libraries

#include <iostream> //Input - Output Library

#include <cstdlib> //Random Number Library

#include <ctime> //Time Library

#include <string> //String Library

#include <iomanip> //Format Library

#include <fstream> //File I/O Library

using namespace std; //Name-space under which system libraries exist

//User Libraries

//Global Constants

int NUMPERC=100;//converting number to percent & vice versa

//Function Prototypes

char nxtCard(char[],int,char);

char shwsuit(int [],int);

char shwface(int [],int);

void preFlop(int [],int,float &,float &,float &,float &);

void dealBet(int [],int,float &,float &,float &,float &);

//Execution begins here

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

//Texas Hold'em Rules

fstream inFile;

string rules;

int num=373;

inFile.open("TexasHoldemRules.dat");

for(int count=1;count<=num;count++){

inFile>>rules;

cout<<rules<<" ";

if (count==7||count==27||count==37||count==56||count==66||count==86

||count==101||count==123||count==138||count==162||count==177

||count==233||count==241||count==256||count==266||count==282

||count==294||count==310||count==319||count==328||count==341)

cout<<endl<<endl;

}

inFile.close();

//Note to Player

cout<<"\n\n\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\n\*\t\t\t\t\t\tWARNING!: THIS GAME IS INCOMPLETE, THERE ARE SOME

HANDS THAT "

"HAVE NOT BEEN INCLUDED!!\t\t\t\t\t\t\*"

"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

//Set the random number seed

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

//Declare variables

const int SIZE=52;

int cards[SIZE]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,

21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,

38,39,40,41,42,43,44,45,46,47,48,49,50,51,52};//deck of cards

char choice;//Menu choice

float bet, pot=0.0f, pMoney=500.0f, oMoney=500.0f;//bet related variables

//Loop

cout<<"\n\nWelcome to the Table. This is a low stakes game, so the average "

"pot is usually less than $100. "<<"\nThe buy-in is $"<<pMoney<<"."<<endl;

do{

cout<<"Press 1 if you would like to play."<<endl;

cin>>choice;

switch(choice){

case'1':{

//Output money

pot=0.0f;//resets pot

cout<<"\nYour money = $"<<pMoney<<endl;

cout<<"opps money = $"<<oMoney<<endl;

cout<<"pot money = $"<<pot<<endl;

//Output cards; for test

cout<<"\nThe cards before shuffling: ";

int cntr=0;

do{

cout<<cards[cntr]<<" ";

cntr++;

}while(cntr<SIZE);

cout<<endl;

//Shuffle Cards

for(int i=0;i<7;i++){

for(int j=0;j<52;j++){

int l=rand()%SIZE, temp;

temp=cards[j];

cards[j]=cards[l];

cards[l]=temp;

}

}

//Output cards again; for test

cout<<"The cards after shuffling: ";

int countr=0;

while(countr<SIZE){

cout<<cards[countr]<<" ";

countr++;

}

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

//Pre-Flop (Deal cards to players)

preFlop(cards,SIZE,oMoney,pot,pMoney,bet);

if(replay==true)break;

//Output money

cout<<"Your money = $"<<pMoney<<endl;

cout<<"opps money = $"<<oMoney<<endl;

cout<<"pot money = $"<<pot<<endl;

cout<<"The flop is being dealt..."<<endl<<endl;

//The Flop

//Output Cards

cout<<"Cards in your hand: "<<shwface(cards,1)<<shwsuit(cards,1)<<" "

<<shwface(cards,3)<<shwsuit(cards,3)<<" Community Cards: "

<<shwface(cards,4)<<shwsuit(cards,4)<<" "<<shwface(cards,5)

<<shwsuit(cards,5)<<" "<<shwface(cards,6)<<shwsuit(cards,6)<<endl;

dealBet(cards,SIZE,oMoney,pot,pMoney,bet);

if(replay==true)break;

//Output money

cout<<"Your money = $"<<pMoney<<endl;

cout<<"opps money = $"<<oMoney<<endl;

cout<<"pot money = $"<<pot<<endl;

cout<<"The turn is being dealt... "<<endl<<endl;

//The Turn

//Output Cards

cout<<"Cards in your hand: "<<shwface(cards,1)<<shwsuit(cards,1)<<" "

<<shwface(cards,3)<<shwsuit(cards,3)<<" Community Cards: "

<<shwface(cards,4)<<shwsuit(cards,4)<<" "<<shwface(cards,5)

<<shwsuit(cards,5)<<" "<<shwface(cards,6)<<shwsuit(cards,6)

<<" "<<shwface(cards,7)<<shwsuit(cards,7)<<endl;

dealBet(cards,SIZE,oMoney,pot,pMoney,bet);

//Output money

cout<<"Your money = $"<<pMoney<<endl;

cout<<"opps money = $"<<oMoney<<endl;

cout<<"pot money = $"<<pot<<endl;

cout<<"The river is being dealt... "<<endl<<endl;

//The River

//Output Cards

cout<<"Cards in your hand: "<<shwface(cards,1)<<shwsuit(cards,1)<<" "

<<shwface(cards,3)<<shwsuit(cards,3)<<" Community Cards: "

<<shwface(cards,4)<<shwsuit(cards,4)<<" "<<shwface(cards,5)

<<shwsuit(cards,5)<<" "<<shwface(cards,6)<<shwsuit(cards,6)

<<" "<<shwface(cards,7)<<shwsuit(cards,7)<<" "<<shwface(cards,8)

<<shwsuit(cards,8)<<endl;

dealBet(cards,SIZE,oMoney,pot,pMoney,bet);

//Output money

cout<<"Your money = $"<<pMoney<<endl;

cout<<"opps money = $"<<oMoney<<endl;

cout<<"pot money = $"<<pot<<endl;

cout<<"Reveal your cards... "<<endl<<endl;

//Output all cards

cout<<"Cards in your hand: "<<shwface(cards,1)<<shwsuit(cards,1)

<<" "<<shwface(cards,3)<<shwsuit(cards,3)

<<" Cards in your opponent's hand: "<<shwface(cards,0)

<<shwsuit(cards,0)<<" "<<shwface(cards,2)<<shwsuit(cards,2)

<<" Community Cards: "<<shwface(cards,4)<<shwsuit(cards,4)<<" "

<<shwface(cards,5)<<shwsuit(cards,5)<<" "<<shwface(cards,6)

<<shwsuit(cards,6)<<" "<<shwface(cards,7)<<shwsuit(cards,7)<<" "

<<shwface(cards,8)<<shwsuit(cards,8)<<endl;

//Variables for hand determination (player)

bool highP=true;//high card

bool pairP=false;//pair, 2 cards with same face value

bool toPairP=false;//two pair, 2 different cards with pairs

bool thrKndP=false;//three of a kind, 3 cards with same face value

bool flushP=false;//flush, 5 cards with the same suit

bool flHousP=false;//full house, three of a kind + a pair

bool frKndP=false;//four of a kind, 4 cards with same face value

//Variables for hand determination (opponent)

bool highO=true;//high card

bool pairO=false;//pair, 2 cards with same face value

bool toPairO=false;//two pair, 2 different cards with pairs

bool thrKndO=false;//three of a kind, 3 cards with same face value

bool flushO=false;//flush, 5 cards with the same suit

bool flHousO=false;//full house, three of a kind + a pair

bool frKndO=false;//four of a kind, 4 cards with same face value

//Finding pairs in the opponent's hand

const int COL=3;

int count[13][COL]={};//counter array

if (shwface(cards,0)==shwface(cards,2))count[0][1]+=1;//finds pair in opponent's hole cards

for (int i=4;i<9;i++){//finds pairs with opp hole cards in com

if (shwface(cards,0)==shwface(cards,i))count[0][1]+=1;

}

for (int i=4;i<9;i++){//finds pairs with opp hole cards in com

if (shwface(cards,2)==shwface(cards,i))count[1][1]+=1;

}

for(int i=4;i<9;i++){//finds pairs in com cards

for(int j=i+1;j<9;j++){

if (shwface(cards,i)==shwface(cards,j))count[2][1]+=1;

}

}

//Finding cards with same suit

if (shwsuit(cards,0)==shwsuit(cards,2))count[0][2]+=1;//finds same suit in opponent's hole cards

for (int i=4;i<9;i++){//finds same suit with opp hole cards in com

if (shwsuit(cards,0)==shwsuit(cards,i))count[0][2]+=1;

}

for (int i=4;i<9;i++){//finds same suit with opp hole cards in com

if (shwsuit(cards,2)==shwsuit(cards,i))count[1][2]+=1;

}

for(int i=4;i<9;i++){//finds same suit in com cards

for(int j=i+1;j<9;j++){

if (shwsuit(cards,i)==shwsuit(cards,j))count[2][2]+=1;

}

}

//Determine hand

for(int j=0;j<=2;j++){//pairs

if(count[j][1]==1)pairO=true;

else if(count[j][1]&&count[j+1][1]==1)toPairO=true;

else if(count[j][1]==2)thrKndO=true;

else if(count[j][1]==2&&count[j+1][1]==1)flHousO=true;

else if(count[j][1]==3)frKndO=true;

else highO=true;

}

for(int i=0;i<=2;i++){//flush

if(count[i][2]==5)flushO=true;

}

//Finding pairs in the player's hand

if (shwface(cards,1)==shwface(cards,3))count[3][1]+=1;

for (int i=4;i<9;i++){

if (shwface(cards,1)==shwface(cards,i))count[3][1]+=1;

}

for (int i=4;i<9;i++){

if (shwface(cards,3)==shwface(cards,i))count[4][1]+=1;

}

for(int i=4;i<9;i++){

for(int j=i+1;j<9;j++){

if (shwface(cards,i)==shwface(cards,j))count[5][1]+=1;

}

}

//Finding cards with same suit in player's hand

if (shwsuit(cards,0)==shwsuit(cards,2))count[3][2]+=1;//finds same suit in opponent's hole cards

for (int i=4;i<9;i++){//finds same suit with opp hole cards in com

if (shwsuit(cards,0)==shwsuit(cards,i))count[3][2]+=1;

}

for (int i=4;i<9;i++){//finds same suit with opp hole cards in com

if (shwsuit(cards,2)==shwsuit(cards,i))count[4][2]+=1;

}

for(int i=4;i<9;i++){//finds same suit in com cards

for(int j=i+1;j<9;j++){

if (shwsuit(cards,i)==shwsuit(cards,j))count[5][2]+=1;

}

}

//Determine hand

for(int j=3;j<=5;j++){

if(count[j][1]==1)pairP=true;

else if(count[j][1]&&count[j+1][1]==1)toPairP=true;

else if(count[j][1]==2)thrKndP=true;

else if(count[j][1]==2&&count[j+1][1]==1)flHousP=true;

else if(count[j][1]==3)frKndP=true;

else highP=true;

}

for(int i=3;i<=5;i++){//flush

if(count[i][2]==5)flushP=true;

}

//Output hands of player and opponent

if (pairP==true)cout<<"You have a pair."<<endl<<endl;

else if(toPairP==true)cout<<"You have two pair."<<endl<<endl;

else if(thrKndP==true)cout<<"You have three of a kind."<<endl<<endl;

else if(flHousP==true)cout<<"You have a full house."<<endl<<endl;

else if(frKndP==true)cout<<"You have four of a kind."<<endl<<endl;

else if(flushP==true)cout<<"You have a flush."<<endl<<endl;

else cout<<"You have a high card."<<endl<<endl;

if (pairO==true)cout<<"Your opponent has a pair."<<endl<<endl;

else if(toPairO==true)cout<<"Your opponent has two pair."<<endl<<endl;

else if(thrKndO==true)cout<<"Your opponent has three of a kind."<<endl<<endl;

else if(flHousO==true)cout<<"Your opponent has a full house."<<endl<<endl;

else if(frKndO==true)cout<<"Your opponent has four of a kind."<<endl<<endl;

else if(flushO==true)cout<<"Your opponent has a flush."<<endl<<endl;

else cout<<"Your opponent has a high card."<<endl<<endl;

//Compare hands

if(pairP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with a pair."<<endl;

pMoney+=pot;

}else if(pairO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with a pair."

<<endl;

oMoney+=pot;

}else if(toPairP==true&&pairO==true||toPairP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with a two pair."<<endl;

pMoney+=pot;

}else if(toPairO==true&&pairP==true||toPairO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with a two pair."

<<endl;

oMoney+=pot;

}else if(thrKndP==true&&toPairO==true||thrKndP==true&&pairO==true

||thrKndP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with three of a kind."<<endl;

pMoney+=pot;

}else if(thrKndO==true&&toPairP==true||thrKndO==true&&pairP==true

||thrKndO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with "

"three of a kind."<<endl;

oMoney+=pot;

}else if(flushP==true&&thrKndO==true||flushP==true&&toPairO==true

||flushP==true&&pairO==true||flushP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with a flush."<<endl;

pMoney+=pot;

}else if(flushO==true&&thrKndP==true||flushO==true&&toPairP==true

||flushO==true&&pairP==true||flushO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with a flush."<<endl;

oMoney+=pot;

}else if(flHousP==true&&flushO==true||flHousP==true&&thrKndO==true

||flHousP==true&&toPairO==true||flHousP==true&&pairO==true

||flHousP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with a full house."<<endl;

pMoney+=pot;

}else if(flHousO==true&&flushP==true||flHousO==true&&thrKndP==true

||flHousO==true&&toPairP==true||flHousO==true&&pairP==true

||flHousO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with a full house."

<<endl;

oMoney+=pot;

}else if(frKndP==true&&flHousO==true||frKndP==true&&flushO==true

||frKndP==true&&thrKndO==true||frKndP==true&&toPairO==true

||frKndP==true&&pairO==true||frKndP==true&&highO==true){

cout<<"You won the pot of $"<<pot<<" with four of a kind."<<endl;

pMoney+=pot;

}else if(frKndO==true&&flHousP==true||frKndO==true&&flushP==true

||frKndO==true&&thrKndP==true||frKndO==true&&toPairP==true

||frKndO==true&&pairP==true||frKndO==true&&highP==true){

cout<<"Your opponent won the pot of $"<<pot<<" with four of a "

"kind."<<endl;

oMoney+=pot;

}else{

cout<<"The pot is being split between you and your opponent."<<endl;

pot/=2;

pMoney+=pot;

oMoney+=pot;

}

break;}

default:{

cout<<endl<<"Thank you for playing."<<endl;

if(pMoney>500){

cout<<"You are walking away with $"<<pMoney<<"!"

<<"\nCongratulations!"<<endl;

}else if(pMoney<500){

cout<<"You are walking away with $"<<pMoney<<"."

<<"\nBetter Luck Next Time!"<<endl;

}else{

cout<<"You are walking away with $"<<pMoney

<<"!\nGood Job!"<<endl;

}

}

}

}while(choice=='1');

//Exit stage right!

return 0;

}

void dealBet(int cards[],int size,float &oMoney,float &pot,float &pMoney,float &bet){

string betChce;//call or fold, bet choice

//Deal community cards

cards[4],cards[5],cards[6],cards[7],cards[8];

//Output all visible cards, community cards and player hole cards

//Opponent's bet

bet=rand()%15+1;

oMoney-=bet;

pot+=bet;

cout<<"bet $"<<bet<<endl<<endl;

//Input data for bet

cout<<"Your opponent made a bet of $"<<bet<<". Do you want to call "

"or fold? (Enter \"call\" or \"fold\" to choose)"<<endl;

cin>>betChce;

if(betChce=="call"){

cout<<"You called. The turn is being dealt... "<<endl<<endl;

pMoney-=bet;

pot+=bet;

}else{

cout<<"You folded. Your opponent won the pot of $"<<pot<<". ";

oMoney+=pot;

exit(0);

}

}

void preFlop(int cards[],int size,float &oMoney,float &pot,float &pMoney,float &bet){

string betChce;//call or fold, bet choice

//Opponent's Hole Cards

cards[0];

cards[2];

//Opponent's bet

bet=rand()%25+1;

oMoney-=bet;

pot+=bet;

cout<<"bet $"<<bet<<endl;

//Output Player's hole cards

cout<<"These are the cards in your hand: "<<shwface(cards,1)

<<shwsuit(cards,1)<<" "<<shwface(cards,3)

<<shwsuit(cards,3)<<endl<<endl;

//Input data for bet

cout<<"Your opponent placed an initial bet of $"<<bet<<" into the pot. "

"Do you want to call or fold? (Enter \"call\" or \"fold\" "

"to choose) "<<endl;

cin>>betChce;

if(betChce=="call"){

cout<<"You called your opponent's bet of $"<<bet<<"."<<endl<<endl;

pMoney-=bet;

pot+=bet;

}else{

cout<<"You folded. Your opponent won the pot of $"<<pot;

oMoney+=pot;

exit(0);

}

}

char shwface(int cards[], int i){

switch(cards[i]%13){

case 1:return '2';

case 2:return '3';

case 3:return '4';

case 4:return '5';

case 5:return '6';

case 6:return '7';

case 7:return '8';

case 8:return '9';

case 9:return 'T';

case 10:return 'J';

case 11:return 'Q';

case 12:return 'K';

default:return 'A';

}

}

char shwsuit(int cards[], int i){

if(cards[i]<=13)return 'S';//S -> Spades

if(cards[i]<=26)return 'D';//D -> Diamond

if(cards[i]<=39)return 'C';//C -> Clubs

return 'H';//H -> Hearts;

}