Project 2

Game Title:

BlackJack (21) Card Game

Project Includes:

Versions 4 & 5 of My Development

Course:

CSC 5 Summer 2017

Section:

45549

Date Due:

July 27, 2017

By:

Javier Ventura

Introduction:

Growing up, my family played many board games, card games, and domino's games. I am relatively much younger than my cousins, older siblings,and my parents are older than me by a higher age difference than you would normally see. So naturally, my younger brother and I were interested in video games. However, whenever it came to playing any games with the rest of the family, of course, it was "non-tech" related. My dad really enjoys casino type games. So I wanted to try and make a 21 version with what we know about C++.

Rules:

21 is a pretty simple game in terms of its logic. A player starts with two cards and based on the worth of each card tries to get as close as possible to the number 21. However, any number above 21 is an automatically loss. The player at a casino is asked if he wants another card or not, where they are allowed to say no if they are under 21 and feel it is not a good idea to ask for another card because of the chance of going over 21. Every number in the deck is worth itself, so a 2 is worth 2 and a 9 is worth 9. The letters are all worth 10. An Ace is allowed to be declared as 1 or 11. So if you have a letter and an Ace, you win with 21. However if you have a total of 12 and get an Ace it is better to make the Ace worth 1 to not go over. Usually in a casino it is a couple of players versus a dealer who plays an individual battle against each player. That means players are not so much aiming to beat each other but the dealer. So naturally the player

Summary of Code:

As mentioned above, the logic is not extremely hard. However, using only concepts we have learned in the class that is on the checklist, then the lines did get repetitive. Most of the code is making sure, that the sums are right as the player gets another card. After the player sees their cards then it asks if they want another card and it displays the card along the previous cards. Many lines were used to declare variables because if a 2 was given in card 1 it is worth 2 but if that card would have been drawn as card 2 then it still has to be worth 2. Getting that to match, along with increasing the possible sum of the cards was a lot of lines using if statements.

Total Lines : 960
Blank and Comment Lines: 100
Variable Lines:92
Library Lines:3
Other Lines: 765

Versions 1,2, & 3 from Project 1:

Version 1 was mostly just getting the program to run,giving out the cards, yet it is still playable considering that the basic rules are all there. Version one was just getting the two cards and asking for more. So if you go over 21, you lose, if you get 21 you win automatically,. The exception is staying under 21, because there is no one to compare against.

Version 2 is a bit more realistic with adding bets and a dealer. At the beginning of the round, the screen prompts the player to enter a bet for the round. If the player wins then their bet is doubled since the dealer will match their bet, just like in a casino. If they lose that round, then their bet is taken by the dealer. The player has his turns first, meaning the dealer's hand is not shown until the player reaches 4 cards or does not want another card. Again, this is similar to a typical casino where the dealer shows his cards after all the players.

For Version 3 I again attempted at making the game a bit more realistic and not limiting the game to one round. As most people stay at a table for multiple rounds and convert much more money into chips, than the worth of just one round of betting. So I added a money to chips conversion, into a total pot. From this total pot you can choose how much you want to put forward for the round and if you lose your bet then that amount is subtracted from your total pot. Likewise if you win then your bet is matched and that extra is added to your pot. The bet per round is also checked to make sure you don’t put any amount above what is in your pot. I also added multiple rounds, so you can play without having to rerun the program. The game ends if your pot is 0 or you do not want to play any more.

Version 4 & 5:

Versions 4 and 5 were additions to the project that helped improve how smooth it ran. I allowed the dealer to get another a card if their sum was below 17. This follows the rules of a dealer in a real game of 21. The rest of the additions to the project were using arrays, vectors and searching methods to add prizes if the player reaches a certain amount of wins. The Display of the score was also added, although casinos wouldn’t do this, it is a good check for the player to see their progression throughout the rounds.

Possible Improvements:

There is a lot of room for the project overall and I have much practice to do. In a real casino, when you play and your two initial cards are the same you are allowed to split, put a bet on each card. This is allowing you to play twice in a round. This would be something figure out for the project .The player is only allowed to ask for 2 cards more than the initial ones given. In BlackJack sometimes this is not a big deal since most games end early but for the exceptions where this is not the case then it is important to add more cards. Another idea is for more players to be involved to simulate an actual table. Adding better graphics is another option to make it seem more like an actual simulator or for people who really like to play casino type games on their phones or computers at home. I know Dr. Lehr talked about the next courses in the C++ series so maybe after taking those I can further develop this project along with others.

Total Lines: 1284
Variable Lines: 111
Blank Lines: 95
Actual Lines: 1078

Pseudo Code:

Include libraries
Declare variables
Input total pot
Input bet for first round
If bet is larger ask for bet again
Give user first two cards
If card one or two is an ace
    Ask user to input if they want that card to be worth 1 or 11
If neither is an ace,
    values are initialized to zero
If card one or two is a two,
    the worth for that card is two
If neither is a two,
    the worth of both cards are initialized to zero
If card one or two is a three,
    the worth for that card is three
If neither is a three,
    the worth of both cards are initialized to zero
If card one or two is a four,
    the worth for that card is four
If neither is a four,
    the worth of both cards are initialized to zero
If card one or two is a five,
    the worth for that card is five
If neither is a five,
    the worth of both cards are initialized to zero
If card one or two is a six,
    the worth for that card is six
If neither is a six,
    the worth of both cards are initialized to zero
If card one or two is a seven,
    the worth for that card is seven
If neither is a seven,
    the worth of both cards are initialized to zero
If card one or two is an eight,
    the worth for that card is an eight
If neither is an eight,
    the worth of both cards are initialized to zero
If card one or two is a nine,
    the worth for that card is nine
If neither is a nine,
    the worth of both cards are initialized to zero

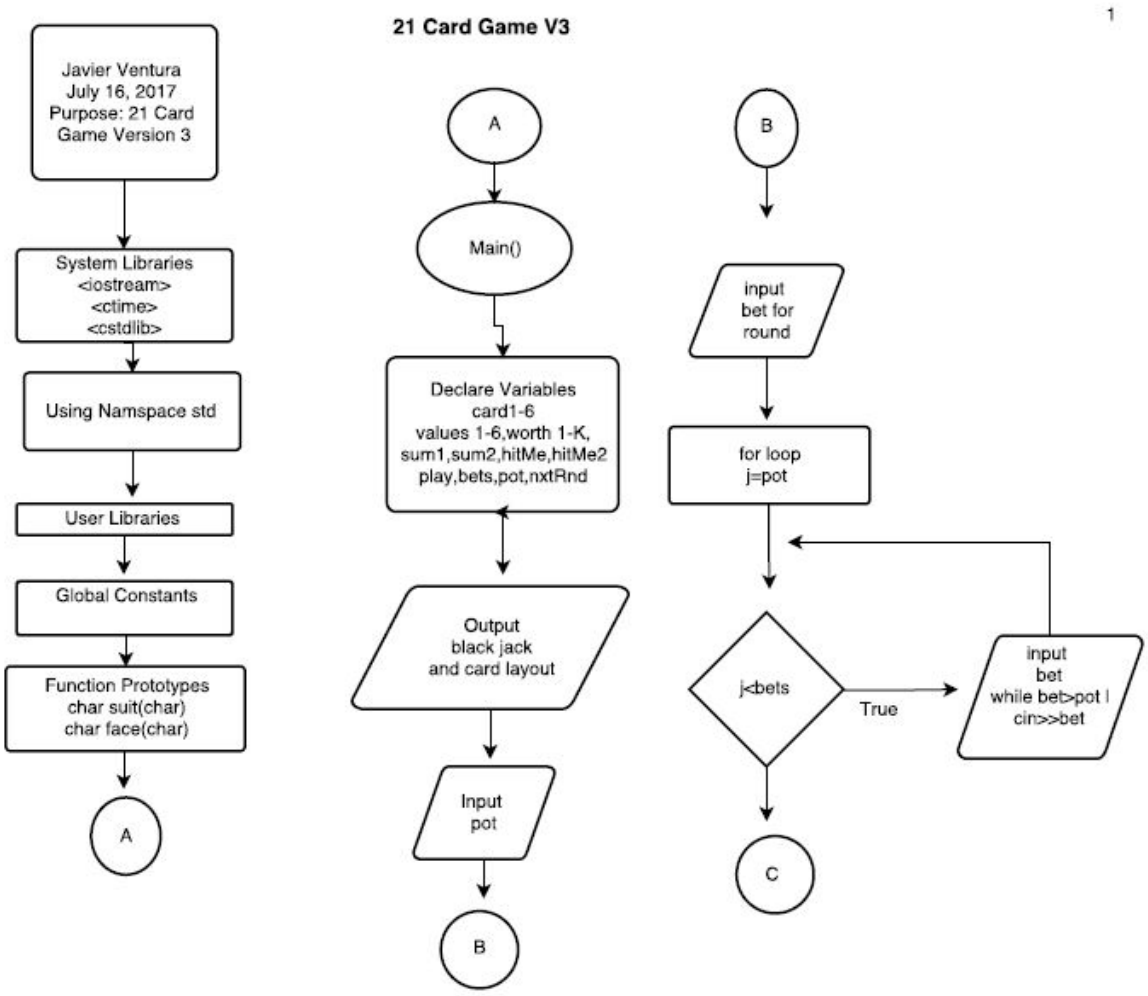
If card one or two is a ten,
    the worth for that card is ten
If neither is a ten,
    the worth of both cards are initialized to zero
If card one or two is a jack,
    the worth for that card is ten
If neither is a jack,
    the worth of both cards are initialized to zero
If card one or two is a queen,
    the worth for that card is ten
If neither is a queen,
    the worth of both cards are initialized to zero
If card one or two is a king,
    the worth for that card is ten
If neither is a king,
    the worth of both cards are initialized to zero
The sum is all values added together
Display sum
If it is 21,
    Player wins, bet is added back to pot
If it is greater than 21
    Player loses, bet is subtracted from pot,
    If pot is 0
        Player can no longer play
If sum is still below 21
    Prompt user wants if they want another card
If No
    Skip to dealers cards
If Yes
    Display last two cards and new one, with new total
If it is 21,
    Player wins, bet is added back to pot
If it is greater than 21
    Player loses, bet is subtracted from pot,
    If pot is 0
        Player can no longer play
If sum is still below 21
    Prompt user wants if they want another card
If No
    Skip to dealers cards
If Yes
    Display last three cards and new one, with new total
If it is 21,
    Player wins, bet is added back to pot

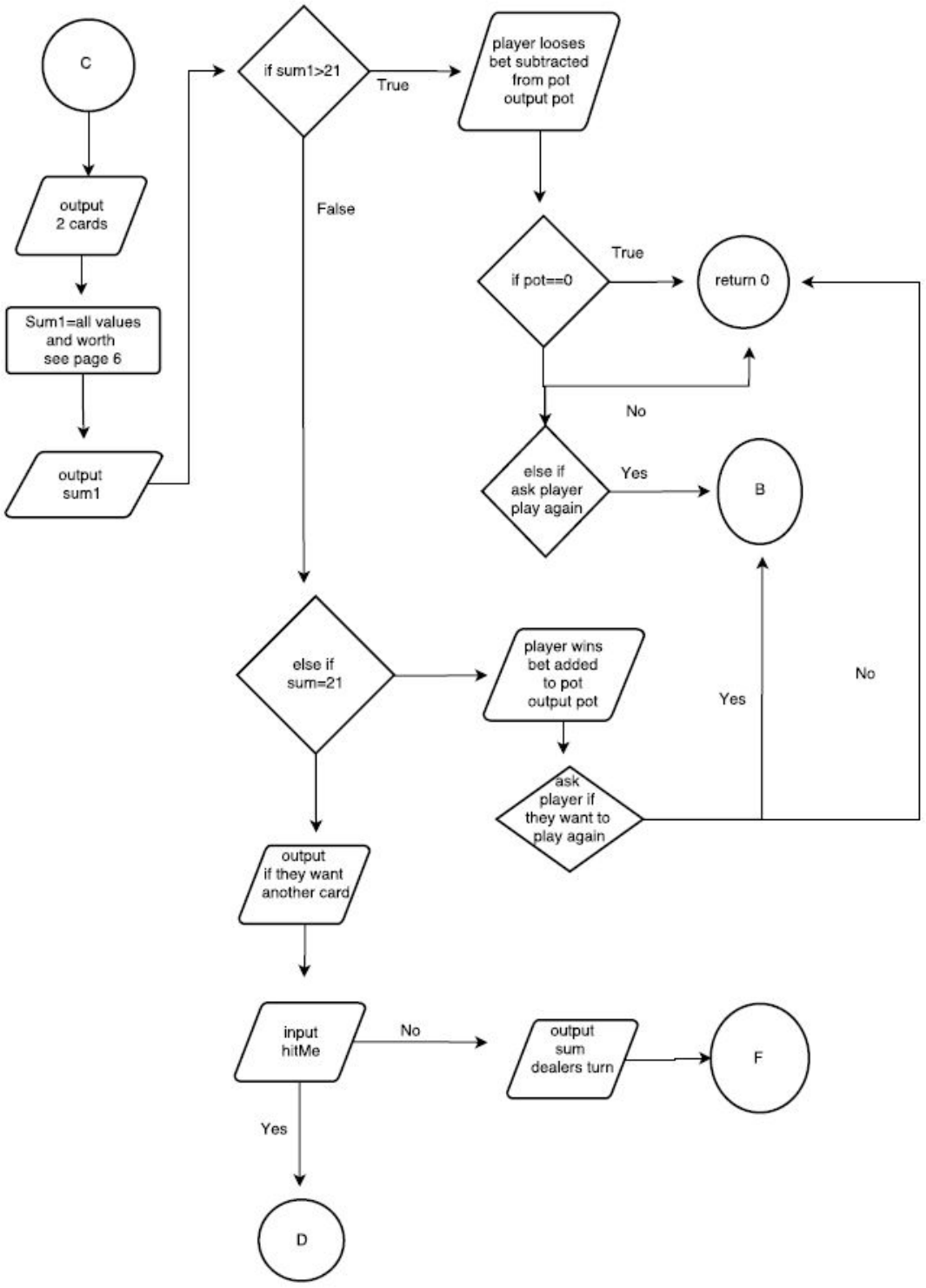
If it is greater than 21
    Player loses, bet is subtracted from pot,
    If pot is 0
        Player can no longer play
If sum is still below 21
    Skip to dealers card now
Display Dealer's cards
Display dealer sum
If it is below 17
    Display third dealer card
        If the sum is below 21 and greater than players sum
            Player loses
                If pot is 0 game is over
        If dealer sum is 21
            Player loses
                If pot is 0 game is over
        If dealer sum is greater than 21
            Player Wins
                If wins equals ten then a bonus prize is given
        If dealer sum is less than 21 but lower than players sum
            Player wins
                If wins equals ten then a bonus prize is given
        If dealer sum is the same as players sum
            Round is a tie

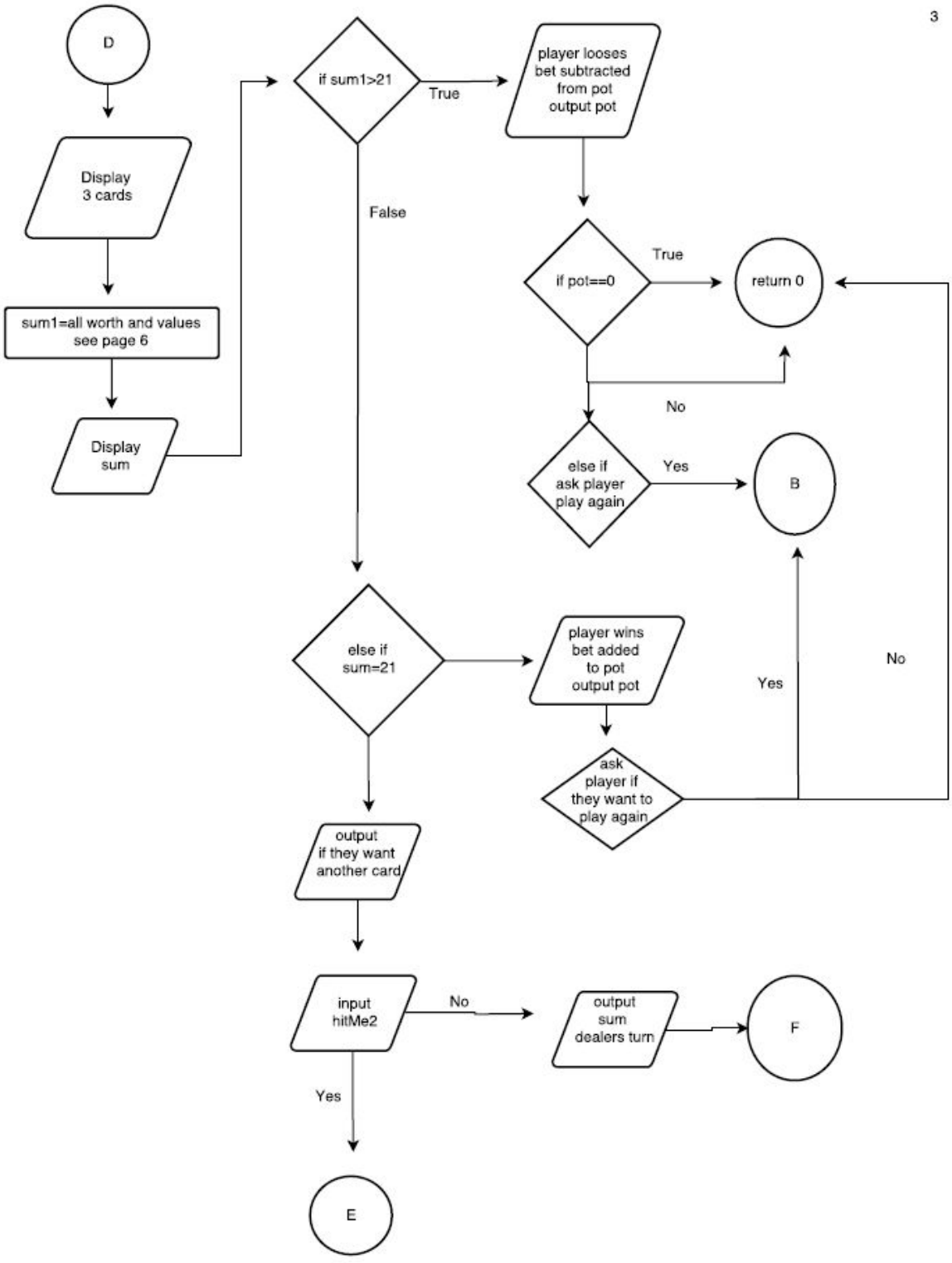
If first two cards sum is between 17 and 21
If first two cards is 21
    Dealer Wins
If first two cards greater than players sum
    Dealers wins
If first two cards less than players sum
    Player wins

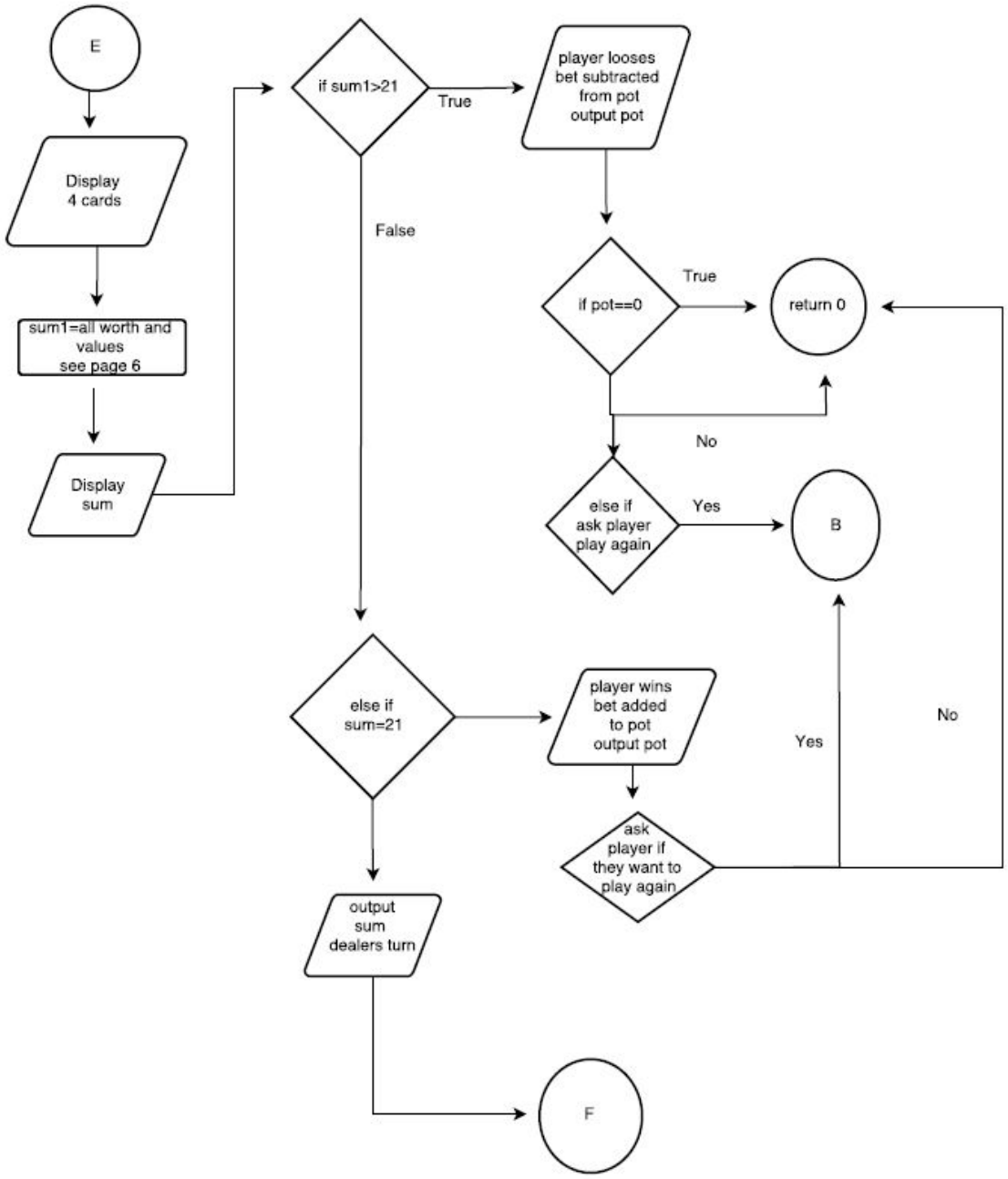
Ask user if they would like to play again
    If Yes
        Return to top of do while loop
    If No
        Game scores are displayed
        Program ends

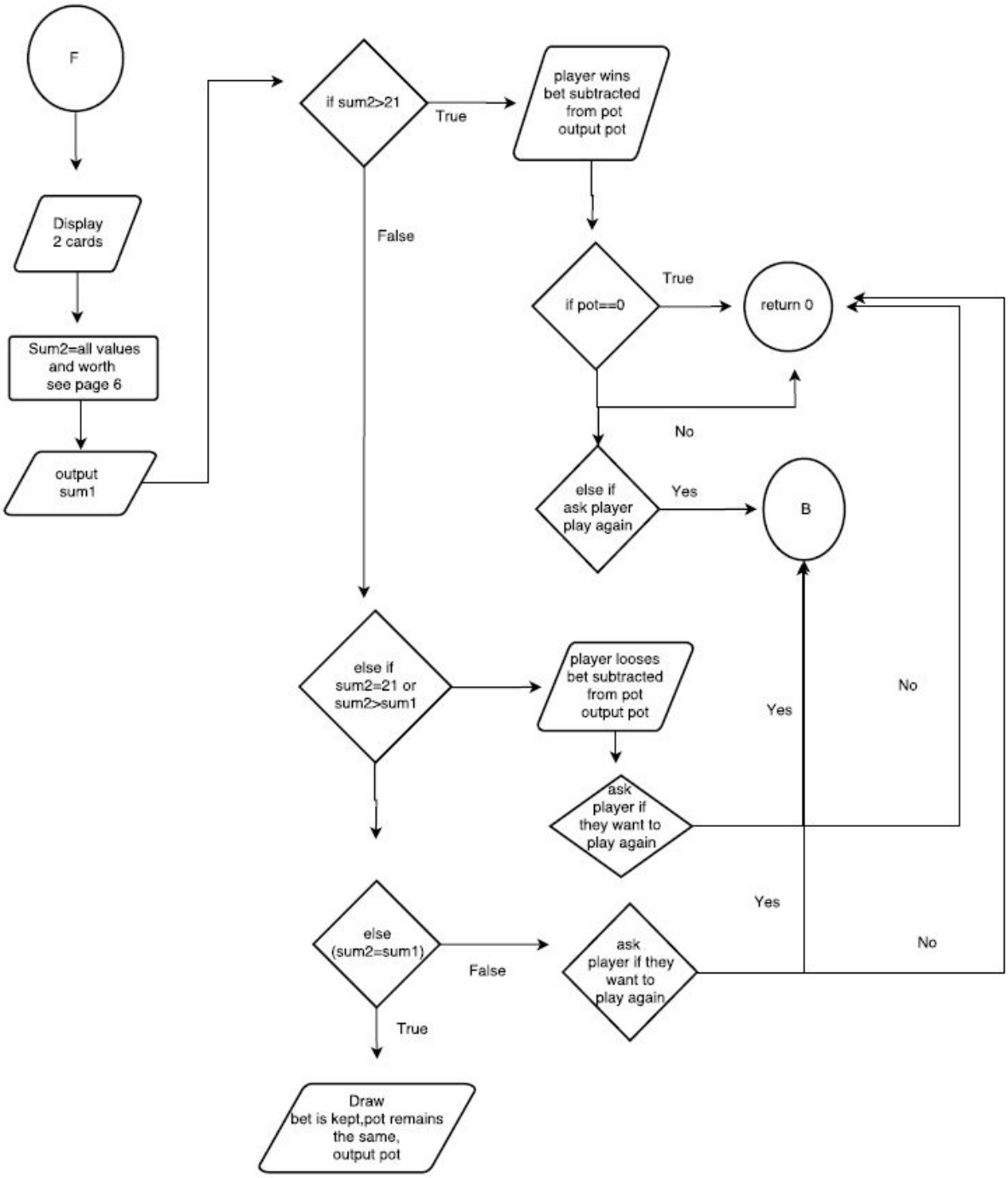
Flowchart:

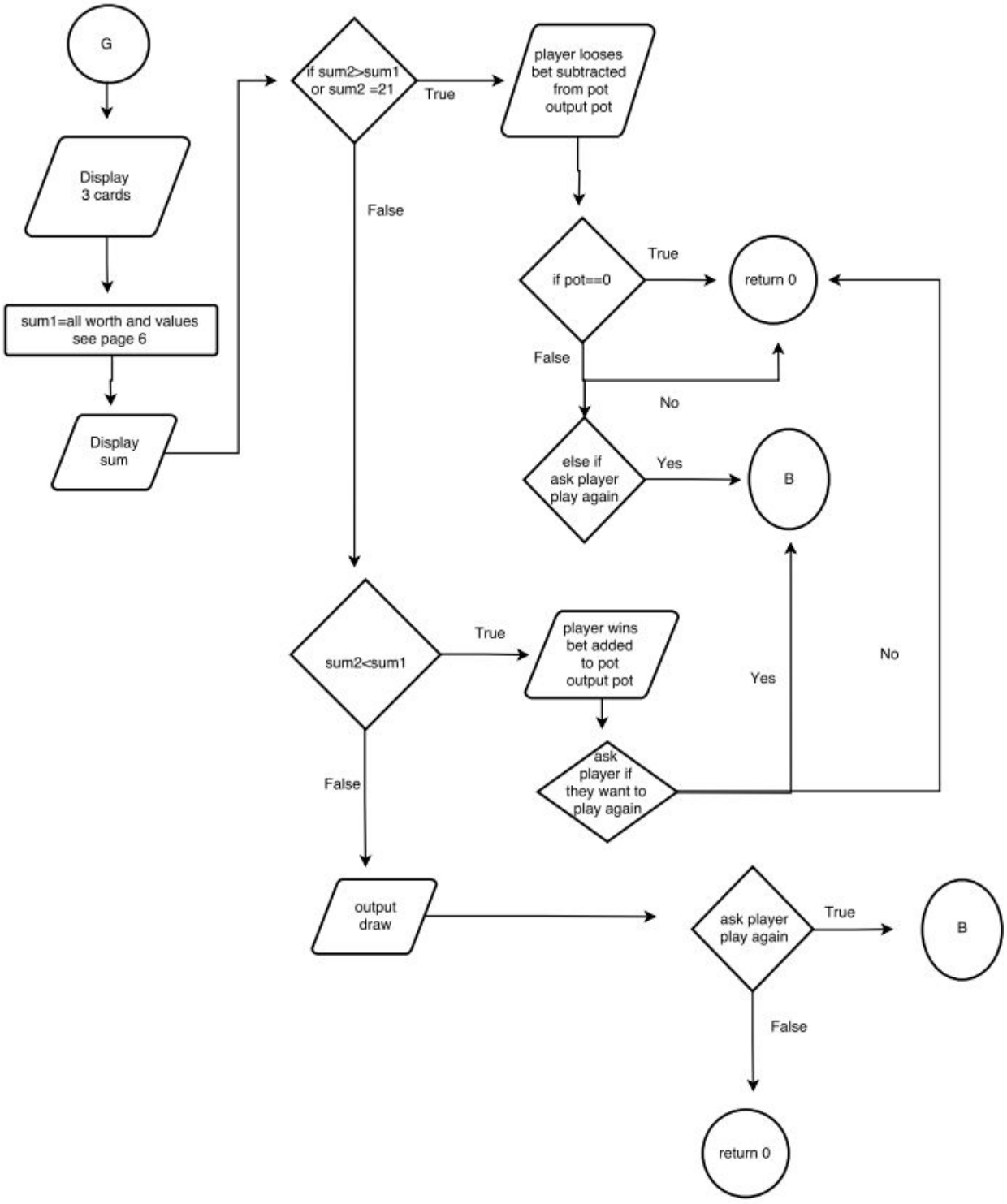

21 Card Game V3
1
Javier Ventura
July 16, 2017
Purpose: 21 Card
Game Version 3
System Libraries
<iostream>
<ctime>
<cstdlib>
Using Namspace std
User Libraries
Global Constants
Function Prototypes
char suit(char)
char face(char)
A
A
Main()
Declare Variables
card1-6
values 1-6,worth 1-K,
sum1,sum2,hitMe,hitMe2
play,bets,pot,nxtRnd
Output
black jack
and card layout
Input
pot
B
B
input
bet for
round
for loop
j=pot
j<bets
True
input
bet
while bet>pot ||
cin>>bet
C

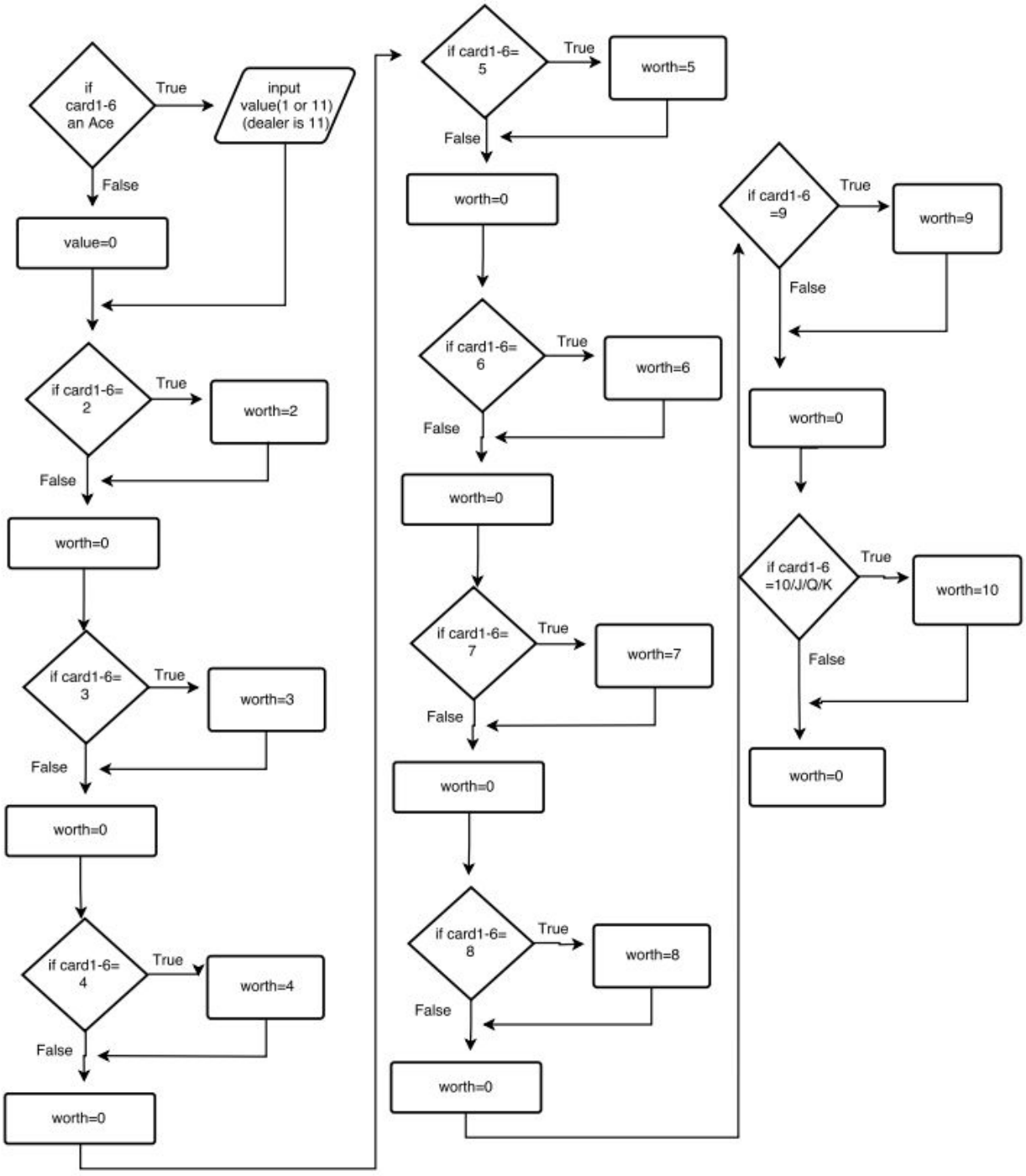

C
output
2 cards
Sum1=all values
and worth
see page 6
output
sum1
if sum1>21
True
False
player looses
bet subtracted
from pot
output pot
if pot==0
True
return 0
No
else if
ask player
play again
Yes
B
else if
sum=21
player wins
bet added
to pot
output pot
ask
player if
they want to
play again
Yes
No
output
if they want
another card
input
hitMe
No
output
sum
dealers turn
F
Yes
D


D
Display
3 cards
sum1=all worth and values
see page 6
Display
sum
if sum1>21
True
player looses
bet subtracted
from pot
output pot
False
if pot==0
True
return 0
No
else if
ask player
play again
Yes
B
else if
sum=21
player wins
bet added
to pot
output pot
ask
player if
they want to
play again
Yes
No
output
if they want
another card
input
hitMe2
No
output
sum
dealers turn
F
Yes
E


E
Display
4 cards
sum1=all worth and
values
see page 6
Display
sum
if sum1>21
True
False
player looses
bet subtracted
from pot
output pot
if pot==0
True
return 0
No
else if
ask player
play again
Yes
B
else if
sum=21
player wins
bet added
to pot
output pot
ask
player if
they want to
play again
Yes
No
output
sum
dealers turn
F


F
Display
2 cards
Sum2=all values
and worth
see page 6
output
sum1
if sum2>21
True
player wins
bet subtracted
from pot
output pot
False
if pot==0
True
return 0
No
else if
ask player
play again
Yes
B
else if
sum2=21 or
sum2>sum1
player looses
bet subtracted
from pot
output pot
Yes
No
ask
player if
they want to
play again
Yes
No
else
(sum2=sum1)
False
ask
player if they
want to
play again
True
Draw
bet is kept,pot remains
the same,
output pot


G
Display
3 cards
sum1=all worth and values
see page 6
Display
sum
if sum2>sum1
or sum2 =21
True
player looses
bet subtracted
from pot
output pot
False
if pot==0
True
return 0
False
No
else if
ask player
play again
Yes
B
sum2<sum1
True
player wins
bet added
to pot
output pot
Yes
No
ask
player if
they want to
play again
False
output
draw
ask player
play again
True
B
False
return 0


if card1-6 an Ace
True
input value(1 or 11) (dealer is 11)
False
value=0
if card1-6= 2
True
worth=2
False
worth=0
if card1-6= 3
True
worth=3
False
worth=0
if card1-6= 4
True
worth=4
False
worth=0
if card1-6= 5
True
worth=5
False
worth=0
if card1-6= 6
True
worth=6
False
worth=0
if card1-6= 7
True
worth=7
False
worth=0
if card1-6= 8
True
worth=8
False
worth=0
if card1-6 =9
True
worth=9
False
worth=0
if card1-6 =10/J/Q/K
True
worth=10
False
worth=0

Code:

```
/*
 * File:   main.cpp
 * Author: Javier Ventura
 * Purpose: 21 Card Game
 * Updates for V4: Dealer gets 3 cards, Bonus Dollar Prizes
 * Created on July 14, 2017, 10:20 PM
 */

//System Libraries
#include <iostream>  //Input - Output Library
#include <cstdlib>   //Random number library
#include <ctime>     //Time library
#include <cmath>     //Math Library
#include <string.h>  //String Library
#include <iomanip>   //Input Output Manipulation
#include <fstream>   //File Input-Output
#include <vector>    //Vector Library

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

//User Libraries

//Global Constants

//Function Prototypes
char suit(char);
char face(char);
void display(int rnd[5],int c[5]);

//Execution begins here
int main(int argc, char** argv){
   ofstream track;

   track.open ("Table.txt");

   //Declare variables
```

```
const int SIZE=7;
string plyr;
string& name=plyr; //Name of player
int value1;     //If card 1 is an Ace, value is either 1 or 11
int value2;     //If card 2 is an Ace, value is either 1 or 11
int value3;     //Possible Value of card 3, 1 or 11 or 0
int value4;     //Possible Value of card 4, 1 or 11 or 0
int value5;     //Possible Value of card 5, 1 or 11 or 0
int value6;     //Possible Value of card 6, 1 or 11 or 0
int value7;     //Possible Value of card 7, 1 or 11 or 0
unsigned int sum1;  //Player Sum
unsigned int sum2;  //Dealer Sum
int worth21; //If Card 1 is a 2 worth is 2,none drawn worth=0
int worth22; //If card 2 is a 2 worth is 2,none drawn worth=0
int worth23; //POssible worth of card 3, 2 or 0
int worth24; //Possible worth of card 4, 2 or 0
int worth25; //Possible worth of card 5, 2 or 0
int worth26; //Possible worth of card 6, 2 or 0
int worth27; //Possible worth of card 7, 2 or 0
int worth31; //If Card 1 is a 3 worth is 3,none drawn worth=0
int worth32; //If card 2 is a 3 worth is 3,none drawn worth=0
int worth33; //Possible worth of card 3, 3 or 0
int worth34; //Possible worth of card 4, 3 or 0
int worth35; //Possible worth of card 5, 3 or 0
int worth36; //Possible worth of card 6, 3 or 0
int worth37; //Possible worth of card 7, 3 or 0
int worth41; //If Card 1 is a 4 worth is 4,none drawn worth=0
int worth42; //If card 2 is a 4 worth is 4,none drawn worth=0
int worth43; //Possible worth of card 3, 4 or 0
int worth44; //Possible worth of card 4, 4 or 0
int worth45; //Possible worth of card 5, 4 or 0
int worth46; //Possible worth of card 6, 4 or 0
int worth47; //Possible worth of card 7, 4 or 0
int worth51; //If Card 1 is a 5 worth is 5,none drawn worth=0
int worth52; //If card 2 is a 5 worth is 5,none drawn worth=0
int worth53; //Possible worth of card 3, 5 or 0
int worth54; //Possible worth of card 4, 5 or 0
int worth55; //Possible worth of card 5, 5 or 0
int worth56; //Possible worth of card 6, 5 or 0
```

```
int worth57; //Possible worth of card 7, 5 or 0
int worth61; //If Card 1 is a 6 worth is 6,none drawn worth=0
int worth62; //If card 2 is a 6 worth is 6,none drawn worth=0
int worth63; //Possible worth of card 3, 6 or 0
int worth64; //Possible worth of card 4, 6 or 0
int worth65; //Possible worth of card 5, 6 or 0
int worth66; //Possible worth of card 6, 6 or 0
int worth67; //Possible worth of card 7, 6 or 0
int worth71; //If Card 1 is a 7 worth is 7,none drawn worth=0
int worth72; //If card 2 is a 7 worth is 7,none drawn worth=0
int worth73; //Possible worth of card 3, 7 or 0
int worth74; //Possible worth of card 4, 7 or 0
int worth75; //Possible worth of card 5, 7 or 0
int worth76; //Possible worth of card 6, 7 or 0
int worth77; //Possible worth of card 7, 7 or 0
int worth81; //If Card 1 is a 8 worth is 8,none drawn worth=0
int worth82; //If card 2 is a 8 worth is 8,none drawn worth=0
int worth83; //Possible worth of card 3, 8 or 0
int worth84; //Possible worth of card 4, 8 or 0
int worth85; //Possible worth of card 5, 8 or 0
int worth86; //Possible worth of card 6, 8 or 0
int worth87; //Possible worth of card 7, 8 or 0
int worth91; //If Card 1 is a 9 worth is 9,none drawn worth=0
int worth92; //If card 2 is a 9 worth is 9,none drawn worth=0
int worth93; //Possible worth of card 3, 9 or 0
int worth94; //Possible worth of card 4, 9 or 0
int worth95; //Possible worth of card 5, 9 or 0
int worth96; //Possible worth of card 6, 9 or 0
int worth97; //Possible worth of card 7, 9 or 0
int worthT1; //If Card 1 is a 10 worth is 10,none drawn worth=0
int worthT2; //If card 2 is a 10 worth is 10,none drawn worth=0
int worthT3; //Possible worth of card 3, 10 or 0
int worthT4; //Possible worth of card 4, 10 or 0
int worthT5; //Possible worth of card 5, 10 or 0
int worthT6; //Possible worth of card 6, 10 or 0
int worthT7; //Possible worth of card 7, 10 or 0
int worthJ1; //If Card 1 is a J worth is 10,none drawn worth=0
int worthJ2; //If card 2 is a J worth is 2,none drawn worth=0
int worthJ3; //Possible worth of card 3, 10 or 0
```

```
   int worthJ4; //Possible worth of card 4, 10 or 0
   int worthJ5; //Possible worth of card 5, 10 or 0
   int worthJ6; //Possible worth of card 6, 10 or 0
   int worthJ7; //Possible worth of card 7, 10 or 0
   int worthQ1; //If Card 1 is a Q worth is 10,none drawn worth=0
   int worthQ2; //If card 2 is a Q worth is 10,none drawn worth=0
   int worthQ3; //Possible worth of card 3, 10 or 0
   int worthQ4; //Possible worth of card 4, 10 or 0
   int worthQ5; //Possible worth of card 5, 10 or 0
   int worthQ6; //Possible worth of card 6, 10 or 0
   int worthQ7; //Possible worth of card 7, 10 or 0
   int worthK1; //If Card 1 is a K worth is 10,none drawn worth=0
   int worthK2; //If card 2 is a K worth is 10,none drawn worth=0
   int worthK3; //Possible worth of card 3, 10 or 0
   int worthK4; //Possible worth of card 4, 10 or 0
   int worthK5; //Possible worth of card 5, 10 or 0
   int worthK6; //Possible worth of card 6, 10 or 0
   int worthK7; //Possible worth of card 7, 10 or 0
   char hitMe;  //The choice to ask for 3rd card
   char hitMe2; //The choice to ask for 4th card
   bool play = true; //Keeps game running, unless players wants to
quit game
   float bets; //Bet per round, must be less than pot
   float pot; //Total pot of money
   char nxtRnd; //The option to play another round
   int a[3]={0,0,0};
   unsigned int bonus[7]={50,100,200,250,300,500,1000};
   vector<int> rnd;
   int wins[5][2]={ {1,2},{3,4},{5,6},{7,8},{9,10}};;


   cout<<"Black Jack"<<endl;
   cout<<"Card Layout: ";
   cout<<"D for Diamonds, S for Spades,C for Clubs,H for
Hearts"<<endl;
   cout<<"T=10 J-Jack Q-Queen K-King A-Ace"<<endl;
   cout<<endl;
   cout<<"Enter Desired User Name: ";
   cin>>plyr;
```

```
cout<<endl;
cout<<"Enter The Amount for your BlackJack Pot (No Commas): $";
cin>>pot;
cout<<endl;

do{
   srand(static_cast<unsigned int>(time(0)));

   char card1=rand()%52+1; //One of the two initial cards for player
   char card2=rand()%52+1; //One of the two initial cards for player
   char card3=rand()%52+1; //Possible extra card for player
   char card4=rand()%52+1; //Possible extra card for player
   char card5=rand()%52+1; //Dealer card 1
   char card6=rand()%52+1; //Dealer card 2
   char card7=rand()%52+1; //Dealer card 3

   int v=rand()%7+0; //[0,6] Pick a random number for bonus prize

   card1!=card2;
   card2!=card3;
   card3!=card4;
   card4!=card5;
   card5!=card6;
   card6!=card7;

   cout<<"Enter Bet for round: $";
   cin>>bets;
   cout<<endl;
   for(int j=pot;j<bets;){
      cout<<"Sorry You don't have that much in your pot."<<setw(4)
            <<" Enter Bet Again: $";
      cin>>bets;
      cout<<endl;
      while(bets>pot){
         cout<<"Sorry You don't have that much in your pot.";
         cout<<" Enter Bet Again: $";
         cin>>bets;
         cout<<endl;
      }
```

```
    }

    cout<<face(card1)<<suit(card1)<<"
"<<face(card2)<<suit(card2)<<endl;

    if(card1==1 || card1==14 || card1==27 || card1==40){
       cout<<endl;
       cout<<"Ace can be 1 or 11."<<endl;
       cin>>value1;
       while(value1!=1&&value1!=11){
          cout<<"1 or 11"<<endl;
          cin>>value1;
          cout<<endl;
       }
    }
    else{
       value1=0;
    }
    if(card2==1 || card2==14 || card2==27 || card2==40){
       cout<<endl;
       cout<<"Ace can be 1 or 11. "<<endl;
       cin>>value2;
       while(value2!=1&&value2!=11){
          cout<<"1 or 11"<<endl;
          cin>>value2;
          cout<<endl;
       }
    }
    else{
       value2=0;
    }

    if(card1==2 || card1==15 || card1==28 || card1==41){
       worth21=2;
    }
    else{
       worth21=0;
    }
    if(card2==2 || card2==15 || card2==28 || card2==41){
```

```
    worth22=2;
}
else{
    worth22=0;
}

if(card1==3 || card1==16 || card1==29 || card1==42){
    worth31=3;
}
else{
    worth31=0;
}

if(card2==3 || card2==16 || card2==29 || card2==42){
    worth32=3;
}
else{
    worth32=0;
}

if(card1==4 || card1==17 || card1==30 || card1==43){
    worth41=4;
}
else{
    worth41=0;
}

if(card2==4 || card2==17 || card2==30 || card2==43){
    worth42=4;
}
else{
    worth42=0;
}

if(card1==5 || card1==18 || card1==31 || card1==44){
    worth51=5;
}
else{
    worth51=0;
```

```
}

if(card2==5 || card2==18 || card2==31 || card2==44){
   worth52=5;
}
else{
   worth52=0;
}

if(card1==6 || card1==19 || card1==32 || card1==45){
   worth61=6;
}
else{
   worth61=0;
}

if(card2==6 || card2==19 || card2==32 || card2==45){
   worth62=6;
}
else{
   worth62=0;
}

if(card1==7 || card1==20 || card1==33 || card1==46){
   worth71=7;
}
else{
   worth71=0;
}

if(card2==7 || card2==20 || card2==33 || card2==46){
   worth72=7;
}
else{
   worth72=0;
}

if(card1==8 || card1==21 || card1==34 || card1==47){
   worth81=8;
```

```
}
else{
   worth81=0;
}

if(card2==8 || card2==21 || card2==34 || card2==47){
   worth82=8;
}
else{
   worth82=0;
}

if(card1==9 || card1==22 || card1==35 || card1==48){
   worth91=9;
}
else{
   worth91=0;
}

if(card2==9 || card2==22 || card2==35 || card2==48){
   worth92=9;
}
else{
   worth92=0;
}

if(card1==10 || card1==23 || card1==36 || card1==49){
   worthT1=10;
}
else{
   worthT1=0;
}

if(card2==10 || card2==23 || card2==36 || card2==49){
   worthT2=10;
}
else{
   worthT2=0;
}
```

```
if(card1==11 || card1==24 || card1==37 || card1==50){
   worthJ1=10;
}
else{
   worthJ1=0;
}

if(card2==11 || card2==24 || card2==37 || card2==50){
   worthJ2=10;
}
else{
   worthJ2=0;
}

if(card1==12 || card1==25 || card1==38 || card1==51){
   worthQ1=10;
}
else{
   worthQ1=0;
}

if(card2==12 || card2==25 || card2==38 || card2==51){
   worthQ2=10;
}
else{
   worthQ2=0;
}


if(card1==13 || card1==26 || card1==39 || card1==52){
   worthK1=10;
}
else{
   worthK1=0;
}

if(card2==13 || card2==26 || card2==39 || card2==52){
   worthK2=10;
```

```
        }
        else{
            worthK2=0;
        }

        //The sum is the worth of two cards drawn, rest are 0

sum1=value1+value2+worth21+worth22+worth31+worth32+worth41+
worth42+worth51

+worth52+worth61+worth62+worth71+worth72+worth81+worth82+wor
th91+worth92+

worthT1+worthT2+worthJ1+worthJ2+worthQ1+worthQ2+worthK1+wor
thK2;
        cout<<"Sum: "<<sum1<<endl;
        cout<<endl;

        //Automatic Win
        if(sum1==21){
            cout<<"21!! You win! "<<endl;
            cout<<endl;
            a[0]++;
            //result(&rnd);
            cout<<"Earnings (This Round): $"<<bets<<endl;
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            if(a[0]==10){
                for(int i=0;i<SIZE;i++){
                    if(bonus[i]==250){
                        cout<<"You have won a bonus prize of: $250";
                        pot=pot+(1.0f*250);
                        cout<<"Your Pot: $"<<pot<<endl;
                    }
                }
            }
        }
        else{
```

```
        cout<<"Do you want another card? Enter Y for Yes,N for
No"<<endl;
        cin>>hitMe;
        cout<<endl;
        //Input Validation
        while(hitMe!='Y' && hitMe!='N'){
           cout<<"Y for Yes and N for No"<<endl;
           cin>>hitMe;
           cout<<endl;
        }
     }

     if(hitMe=='Y'){
        cout<<face(card1)<<suit(card1)<<"
"<<face(card2)<<suit(card2);
        cout<<" "<<face(card3)<<suit(card3)<<endl;

        if(card3==1 || card3==14 || card3==27 || card3==40){
           cout<<endl;
           cout<<"Ace can be 1 or 11."<<endl;
           cin>>value3;
           cout<<endl;
           while(value3!=1 && value3!=11){
              cout<<"1 or 11"<<endl;
              cin>>value3;
              cout<<endl;
           }
        }
        else{
           value3=0;
        }
        if(card3==2 || card3==15 || card3==28 || card3==41){
           worth23=2;
        }
        else{
           worth23=0;
        }
        if(card3==3 || card3==16 || card3==29 || card3==42){
           worth33=3;
```

```
}
else{
   worth33=0;
}

if(card3==4 || card3==17 || card3==30 || card3==43){
   worth43=4;
}
else{
   worth43=0;
}

if(card3==5 || card3==18 || card3==31 || card3==44){
   worth53=5;
}
else{
   worth53=0;
}

if(card3==6 || card3==19 || card3==32 || card3==45){
   worth63=6;
}
else{
   worth63=0;
}

if(card3==7 || card3==20 || card3==33 || card3==46){
   worth73=7;
}
else{
   worth73=0;
}

if(card3==8 || card3==21 || card3==34 || card3==47){
   worth83=8;
}
else{
   worth83=0;
}
```

```
if(card3==9 || card3==22 || card3==35 || card3==48){
   worth93=9;
}
else{
   worth93=0;
}

if(card3==10 || card3==23 || card3==36 || card3==49){
   worthT3=10;
}
else{
   worthT3=0;
}

if(card3==11 || card3==24 || card3==37 || card3==50){
   worthJ3=10;
}
else{
   worthJ3=0;
}
if(card3==12 || card3==25 || card3==38 || card3==51){
   worthQ3=10;
}
else{
   worthQ3=0;
}
if(card3==13 || card3==26 || card3==39 || card3==52){
   worthK3=10;
}
else{
   worthK3=0;
}


sum1=value1+value2+worth21+worth22+worth31+worth32+worth41+
worth42+
```

```
worth51+worth52+worth61+worth62+worth71+worth72+worth81+wort
h82+

worth91+worth92+worthT1+worthT2+worthJ1+worthJ2+worthQ1+wort
hQ2+

worthK1+worthK2+worthK3+worthQ3+worthJ3+worthT3+worth93+wo
rth83+
        worth73+worth63+worth53+worth43+worth33+worth23+value3;

        cout<<"Sum: "<<sum1<<endl;
        cout<<endl;
        if(sum1>21){
           cout<<"You went over 21! Sorry You Lose!"<<endl;
           cout<<"You Lost: $"<<bets<<endl;
           cout<<endl;
           pot=pot-bets;
           cout<<"Your Pot: $"<<showpoint<<pot<<endl;
           cout<<endl;
           a[1]++;
           if(pot==0){
              cout<<"Sorry, Your Pot is empty.Come Back Next
Time"<<endl;
              play=false;
              cout<<endl;
              cout<<plyr<<endl;
              cout<<"---------"<<endl;
              cout<<"Wins: "<<a[0]<<endl;
              cout<<"Losses: "<<a[1]<<endl;
              cout<<"Draws: "<<a[2]<<endl;
              return 0;
           }
        }
        else if(sum1==21){
           cout<<"21!! You win! "<<endl;
           cout<<endl;
           a[0]++;
           cout<<"Earnings (This Round): $"<<bets<<endl;
```

```
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            if(a[0]==10){
               for(int i=0;i<SIZE;i++){
                  if(bonus[i]==250){
                     cout<<"You have won a bonus prize of: $250"<<endl;
                     pot=pot+(1.0f*250);
                     cout<<"Your Pot: $"<<pot<<endl;
                  }
               }
            }
         }

         else{
            cout<<"Do you want another card? Enter Y for Yes,N for
No"<<endl;
            cin>>hitMe2;
            cout<<endl;
            while(hitMe2!='Y' && hitMe2!='N'){
               cout<<"Y for Yes and N for No"<<endl;
               cin>>hitMe2;
               cout<<endl;
            }
         }
         //Overload Function Suit and Face with multiple cards
         if(hitMe2=='Y'){
            cout<<face(card1)<<suit(card1)<<"
"<<face(card2)<<suit(card2);
            cout<<" "<<face(card3)<<suit(card3)<<" "<<face(card4)
                  <<suit(card4)<<endl;

            if(card4==1 || card4==14 || card4==27 || card4==40){
               cout<<endl;
               cout<<"Ace can be 1 or 11."<<endl;
               cin>>value4;
               cout<<endl;
               while(value4!=1&&value4!=11){
                  cout<<"1 or 11"<<endl;
```

```
        cin>>value4;
        cout<<endl;
    }
}
else{
    value4=0;
}
if(card4==2 || card4==15 || card4==28 || card4==41){
    worth24=2;
}
else{
    worth24=0;
}
if(card4==3 || card4==16 || card4==29 || card4==42){
    worth34=3;
}
else{
    worth34=0;
}

if(card4==4 || card4==17 || card4==30 || card4==43){
    worth44=4;
}
else{
    worth44=0;
}

if(card4==5 || card4==18 || card4==31 || card4==44){
    worth54=5;
}
else{
    worth54=0;
}

if(card4==6 || card4==19 || card4==32 || card4==45){
    worth64=6;
}
else{
    worth64=0;
```

```
}

if(card4==7 || card4==20 || card4==33 || card4==46){
   worth74=7;
}
else{
   worth74=0;
}

if(card4==8 || card4==21 || card4==34 || card4==47){
   worth84=8;
}
else{
   worth84=0;
}

if(card4==9 || card4==22 || card4==35 || card4==48){
   worth94=9;
}
else{
   worth94=0;
}

if(card4==10 || card4==23 || card4==36 || card4==49){
   worthT4=10;
}
else{
   worthT4=0;
}

if(card4==11 || card4==24 || card4==37 || card4==50){
   worthJ4=10;
}
else{
   worthJ4=0;
}
if(card4==12 || card4==25 || card4==38 || card4==51){
   worthQ4=10;
}
```

```
            else{
               worthQ4=0;
            }
            if(card4==13 || card4==26 || card4==39 || card4==52){
               worthK4=10;
            }
            else{
               worthK4=0;
            }


sum1=value1+value2+worth21+worth22+worth31+worth32+worth41+
worth42+

worth51+worth52+worth61+worth62+worth71+worth72+worth81+wort
h82+

worth91+worth92+worthT1+worthT2+worthJ1+worthJ2+worthQ1+wort
hQ2+

worthK1+worthK2+worthK3+worthQ3+worthJ3+worthT3+worth93+wo
rth83+

worth73+worth63+worth53+worth43+worth33+worth23+value3+value4
+

worth24+worth34+worth44+worth54+worth64+worth74+worth84+wort
h94+
            worthT4+worthJ4+worthQ4+worthK4;

            cout<<"Sum: "<<sum1<<endl;
            cout<<endl;

            if(sum1>21){
               cout<<"You went over 21! Sorry You Lose!"<<endl;
               cout<<endl;
               cout<<"You Lost: $0"<<endl;
               a[1]++;
               pot=pot-bets;
```

```
            cout<<"Your Pot: $"<<pot<<endl;
            if(pot==0){
               cout<<"Sorry, Your Pot is empty.Come Back Next
Time";
               cout<<endl;
               play=false;
               cout<<endl;
               cout<<plyr<<endl;
               cout<<"---------"<<endl;
               cout<<"Wins: "<<a[0]<<endl;
               cout<<"Losses: "<<a[1]<<endl;
               cout<<"Draws: "<<a[2]<<endl;
               return 0;
            }
         }
         else if(sum1==21){
            cout<<"21!! You Win!!"<<endl;
            cout<<endl;
            a[0]++;
            cout<<"Earnings (This Round): $"<<bets<<endl;
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            if(a[0]==10){
               for(int i=0;i<SIZE;i++){
                  if(bonus[i]==250){
                     cout<<"You have won a bonus prize of:
$250"<<endl;
                     pot=pot+(1.0f*250);
                     cout<<"Your Pot: $"<<pot<<endl;
                  }
               }
            }
         }
         else{
            cout<<"You stayed at: "<<sum1<<endl;
            cout<<endl;
            cout<<"Dealers Turn"<<endl;
            cout<<endl;
```

```
            }
        }
        else if(hitMe2=='N'){
            cout<<endl;
            cout<<"You stayed at: "<<sum1<<endl;
            cout<<endl;
            cout<<"Dealers Turn"<<endl;
            cout<<endl;
        }
    }
    else if(hitMe=='N'){
        cout<<endl;
        cout<<"You stayed at: "<<sum1<<endl;
        cout<<endl;
        cout<<"Dealers Turn"<<endl;
        cout<<endl;
    }

    if(sum1<21){
        cout<<"Dealers Cards:"<<endl;
        cout<<face(card5)<<suit(card5)<<"
"<<face(card6)<<suit(card6)<<endl;

        if(card5==1 || card5==14 || card5==27 || card5==40){
            cout<<endl;
            cout<<"Ace can be 1 or 11."<<endl;
            cout<<"11"<<endl;
            cout<<endl;
            value5=11;
        }
        else{
            value5=0;
        }
        if(card6==1 || card6==14 || card6==27 || card6==40){
            cout<<endl;
            cout<<"Ace can be 1 or 11. "<<endl;
            cout<<"11"<<endl;
            cout<<endl;
            value6=11;
```

```
}
else{
   value6=0;
}

if(card5==2 || card5==15 || card5==28 || card5==41){
   worth25=2;
}
else{
   worth25=0;
}

if(card6==2 || card6==15 || card6==28 || card6==41){
   worth26=2;
}
else{
   worth26=0;
}

if(card5==3 || card5==16 || card5==29 || card5==42){
   worth35=3;
}
else{
   worth35=0;
}

if(card6==3 || card6==16 || card6==29 || card6==42){
   worth36=3;
}
else{
   worth36=0;
}

if(card5==4 || card5==17 || card5==30 || card5==43){
   worth45=4;
}
else{
   worth45=0;
}
```

```
if(card6==4 || card6==17 || card6==30 || card6==43){
   worth46=4;
}
else{
   worth46=0;
}

if(card5==5 || card5==18 || card5==31 || card5==44){
   worth55=5;
}
else{
   worth55=0;
}

if(card6==5 || card6==18 || card6==31 || card6==44){
   worth56=5;
}
else{
   worth56=0;
}

if(card5==6 || card5==19 || card5==32 || card5==45){
   worth65=6;
}
else{
   worth65=0;
}

if(card6==6 || card6==19 || card6==32 || card6==45){
   worth66=6;
}
else{
   worth66=0;
}

if(card5==7 || card5==20 || card5==33 || card5==46){
   worth75=7;
}
```

```
else{
   worth75=0;
}

if(card6==7 || card6==20 || card6==33 || card6==46){
   worth76=7;
}
else{
   worth76=0;
}

if(card5==8 || card5==21 || card5==34 || card5==47){
   worth85=8;
}
else{
   worth85=0;
}

if(card6==8 || card6==21 || card6==34 || card6==47){
   worth86=8;
}
else{
   worth86=0;
}

if(card5==9 || card5==22 || card5==35 || card5==48){
   worth95=9;
}
else{
   worth95=0;
}

if(card6==9 || card6==22 || card6==35 || card6==48){
   worth96=9.0f;
}
else{
   worth96=0;
}
```

```
if(card5==10 || card5==23 || card5==36 || card5==49){
   worthT5=10.0f;
}
else{
   worthT5=0;
}

if(card6==10 || card6==23 || card6==36 || card6==49){
   worthT6=10.0f;
}
else{
   worthT6=0;
}

if(card5==11 || card5==24 || card5==37 || card5==50){
   worthJ5=10.0f;
}
else{
   worthJ5=0;
}

if(card6==11 || card6==24 || card6==37 || card6==50){
   worthJ6=10.0f;
}
else{
   worthJ6=0;
}

if(card5==12 || card5==25 || card5==38 || card5==51){
   worthQ5=10.0f;
}
else{
   worthQ5=0;
}

if(card6==12 || card6==25 || card6==38 || card6==51){
   worthQ6=10.0f;
}
else{
```

```
            worthQ6=0;
        }


        if(card5==13 || card5==26 || card5==39 || card5==52){
            worthK5=10.0f;
        }
        else{
            worthK5=0;
        }

        if(card6==13 || card6==26 || card6==39 || card6==52){
            worthK6=10.0f;
        }
        else{
            worthK6=0;
        }


sum2=value5+value6+worth25+worth26+worth35+worth36+worth45+
worth46+

worth55+worth56+worth65+worth66+worth75+worth76+worth85+wort
h86+

worth95+worth96+worthT5+worthT6+worthJ5+worthJ6+worthQ5+wort
hQ6+
        worthK5+worthK6;

        cout<<"Dealer Sum: "<<sum2<<endl;

        if(sum2<17){
            cout<<endl;
            cout<<face(card5)<<suit(card5)<<"
"<<face(card6)<<suit(card6);
            cout<<" "<<face(card7)<<suit(card7)<<endl;

            if(card7==1 || card7==14 || card7==27 || card7==40){
                cout<<endl;
```

```
    cout<<"Ace can be 1 or 11."<<endl;
    cout<<"11"<<endl;
    value7=11;
    cout<<endl;
}
else{
    value7=0;
}
if(card7==2 || card7==15 || card7==28 || card7==41){
    worth27=2;
}
else{
    worth27=0;
}
if(card7==3 || card7==16 || card7==29 || card7==42){
    worth37=3;
}
else{
    worth37=0;
}

if(card7==4 || card7==17 || card7==30 || card7==43){
    worth47=4;
}
else{
    worth47=0;
}

if(card7==5 || card7==18 || card7==31 || card7==44){
    worth57=2;
}
else{
    worth57=0;
}

if(card7==6 || card7==19 || card7==32 || card7==45){
    worth67=6;
}
else{
```

```
    worth67=0;
}

if(card7==7 || card7==20 || card7==33 || card7==46){
    worth77=7;
}
else{
    worth77=0;
}

if(card7==8 || card7==21 || card7==34 || card7==47){
    worth87=8;
}
else{
    worth87=0;
}

if(card7==9 || card7==22 || card7==35 || card7==48){
    worth97=9;
}
else{
    worth97=0;
}

if(card7==10 || card7==23 || card7==36 || card7==49){
    worthT7=10;
}
else{
    worthT7=0;
}

if(card7==11 || card7==24 || card7==37 || card7==50){
    worthJ7=10;
}
else{
    worthJ7=0;
}
if(card7==12 || card7==25 || card7==38 || card7==51){
    worthQ7=10;
```

```
            }
            else{
               worthQ7=0;
            }
            if(card7==13 || card7==26 || card7==39 || card7==52){
               worthK7=10;
            }
            else{
               worthK7=0;
            }


sum2=value5+value6+value7+worth25+worth26+worth27+worth35+

worth36+worth37+worth45+worth46+worth47+worth55+worth56+wort
h57+

worth65+worth66+worth67+worth75+worth76+worth77+worth85+wort
h86+

worth87+worth95+worth96+worth97+worthT5+worthT6+worthT7+wort
hJ5+

worthJ6+worthJ7+worthQ5+worthQ6+worthQ7+worthK5+worthK6+wo
rthK7;

            cout<<"Dealer Sum: "<<sum2<<endl;
            cout<<endl;

            if(sum2==21){
               cout<<endl;
               cout<<"Dealer Wins!"<<endl;
               cout<<endl;
               cout<<"You Lost: $"<<bets<<endl;
               pot=pot-bets;
               cout<<"Your Pot: $"<<showpoint<<pot<<endl;
               cout<<endl;
               a[1]++;
               if(pot==0){
```

```
                cout<<"Sorry, Your Pot is empty.Come Back Next
Time";
                cout<<endl;
                play=false;
                cout<<endl;
                cout<<plyr<<endl;
                cout<<"---------"<<endl;
                cout<<"Wins: "<<a[0]<<endl;
                cout<<"Losses: "<<a[1]<<endl;
                cout<<"Draws: "<<a[2]<<endl;
                return 0;
             }
          }
          else if(sum2<21 && sum2>sum1){
             cout<<endl;
             cout<<"Dealer Wins!"<<endl;
             cout<<endl;
             cout<<"You Lost: $"<<bets<<endl;
             pot=pot-bets;
             cout<<"Your Pot: $"<<showpoint<<pot<<endl;
             cout<<endl;
             a[1]++;
             if(pot==0){
                cout<<"Sorry, Your Pot is empty.Come Back Next
Time";
                cout<<endl;
                play=false;
                cout<<endl;
                cout<<plyr<<endl;
                cout<<"---------"<<endl;
                cout<<"Wins: "<<a[0]<<endl;
                cout<<"Losses: "<<a[1]<<endl;
                cout<<"Draws: "<<a[2]<<endl;
                return 0;
             }
          }
          else if(sum2>21){
             cout<<endl;
             cout<<"Player Wins!"<<endl;
```

```
            cout<<endl;
            a[0]++;
            cout<<"Earnings (This Round): $"<<bets<<endl;
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            if(a[0]==10){
               for(int i=0;i<SIZE;i++){
                  if(bonus[i]==250){
                     cout<<"You have won a bonus prize of:
$250"<<endl;
                     pot=pot+(1.0f*250);
                     cout<<"Your Pot: $"<<pot<<endl;
                  }
               }
            }
         }
         else if(sum2<21 && sum1>sum2){
            cout<<endl;
            cout<<"Player Wins!"<<endl;
            cout<<endl;
            a[0]++;
            cout<<"Earnings (This Round): $"<<bets<<endl;
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            if(a[0]==10){
               for(int i=0;i<SIZE;i++){
                  if(bonus[i]==250){
                     cout<<"You have won a bonus prize of:
$250"<<endl;
                     pot=pot+(1.0f*250);
                     cout<<"Your Pot: $"<<pot<<endl;
                  }
               }
            }
         }
         else if(sum2=sum1){
            cout<<"Draw This Round"<<endl;
```

```
            cout<<endl;
            pot=pot;
            a[2]++;
            cout<<"Your Pot: $"<<pot<<endl;
            cout<<endl;
         }
      }
      else if(sum2>17 && sum2<21){
         if(sum1>sum2){
            cout<<endl;
            cout<<"Player Wins!"<<endl;
            cout<<endl;
            a[0]++;
            cout<<"Earnings (This Round): $"<<bets<<endl;
            pot=pot+bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            //Linear Search
            if(a[0]==10){
               for(int i=0;i<SIZE;i++){
                  if(bonus[i]==250){
                     cout<<"You have won a bonus prize of:
$250"<<endl;
                     pot=pot+(1.0f*250);
                     cout<<"Your Pot: $"<<pot<<endl;
                  }
               }
            }
         }
         else if(sum2>sum1){
            cout<<endl;
            cout<<"Dealer Wins!"<<endl;
            cout<<endl;
            cout<<"You Lost: $"<<showpoint<<bets<<endl;
            pot=pot-bets;
            cout<<"Your Pot: $"<<showpoint<<pot<<endl;
            cout<<endl;
            a[1]++;
            if(pot==0){
```

```
                cout<<"Sorry, Your Pot is empty.Come Back Next
Time";
                cout<<endl;
                play=false;
                cout<<endl;
                cout<<plyr<<endl;
                cout<<"---------"<<endl;
                cout<<"Wins: "<<a[0]<<endl;
                cout<<"Losses: "<<a[1]<<endl;
                cout<<"Draws: "<<a[2]<<endl;
                return 0;
              }
            }
            else if(sum2==sum1){
               cout<<endl;
               cout<<"Draw This Round"<<endl;
               cout<<endl;
               pot=pot;
               a[2]++;
               cout<<"Your Pot: $"<<pot<<endl;
               cout<<endl;
            }
          }
          else if(sum2==21){
             cout<<endl;
             cout<<"Dealer Wins!"<<endl;
             cout<<endl;
             cout<<"You Lost: $"<<showpoint<<bets<<endl;
             pot=pot-bets;
             cout<<"Your Pot: $"<<showpoint<<pot<<endl;
             cout<<endl;
             a[1]++;
             if(pot==0){
                cout<<"Sorry, Your Pot is empty.Come Back Next Time";
                cout<<endl;
                play=false;
                cout<<endl;
                cout<<plyr<<endl;
                cout<<"---------"<<endl;
```

```
                cout<<"Wins: "<<a[0]<<endl;
                cout<<"Losses: "<<a[1]<<endl;
                cout<<"Draws: "<<a[2]<<endl;
                return 0;
            }
        }
    }
    cout<<"Play again? Y For Yes and N For No"<<endl;
    cin>>nxtRnd;
    if(nxtRnd=='Y'){
        play=true;
    }
    else{
        play=false;
        cout<<endl;
        cout<<plyr<<endl;
        cout<<"---------"<<endl;
        cout<<"Wins: "<<a[0]<<endl;
        cout<<"Losses: "<<a[1]<<endl;
        cout<<"Draws: "<<a[2]<<endl;
        break;
    }
  }while(play=true);

  track.close();

  //Exit stage right!
  return 0;

}

char face(char card){
  switch(card%13){
    case 1:return 'A';
    case 2:return '2';
    case 3:return '3';
    case 4:return '4';
    case 5:return '5';
    case 6:return '6';
```

```
        case 7:return '7';
        case 8:return '8';
        case 9:return '9';
        case 10:return 'T';
        case 11:return 'J';
        case 12:return 'Q';
        default:return 'K';
    }
}

char suit(char card){
    if(card<=13)return 'S';//S -> Spades
    if(card<=26)return 'D';//D -> Diamond
    if(card<=39)return 'C';//C -> Clubs
    return 'H';//H -> Hearts;
}

void display(int rnd[5],int c[5]){
    for(int i=0;i<4;i++){
        cout<<"Round "<<i+1<<": "<<rnd[i]<<" "<<c[i]<<endl;
    }
}
```

Checklist:

Chapter	Section	Topic	Where in Code Line number
2	2	cout	140
	3	libraries	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	32-138
	5	Identifiers	154-160
	6	Integers	110
	7	Characters	154-160
	8	Strings	33
	9	Floats No Doubles	130
	10	Bools	129
	11	Sizeof *****	
	12	Variables 7 characters or less	90
	13	Scope ***** No Global Variables	
	14	Arithmetic operators	382
	15	Comments 20%+	1210
	16	Named Constants	32
	17	Programming Style ***** Emulate	
3	1	cin	169
	2	Math Expression	382
	3	Mixing data types ****	
	4	Overflow/Underflow ****	
	5	Type Casting	149
	6	Multiple assignment *****	
	7	Formatting output	1138
	8	Strings	33
	9	Math Library	13
	10	Hand tracing ******	
4	1	Relational Operators	184
	2	if	1207
	4	If-else	581 & 592
	5	Nesting	576 & 581 & 687
	6	If-else-if	687 707
	7	Flags *****	
	8	Logical operators	1100
	11	Validating user input	169/174
	13	Conditional Operator	1090
	14	Switch	1189

5	1	Increment/Decrement	1158
	2	While	174
	5	Do-while	148
	6	For loop	169
	11	Files input/output both	32
	12	No breaks in loops ******	
6	3	Function Prototypes	25
	5	Passing by value	1188
	8	Returning values from functions	1191
	9	Returning a boolean ******	
	10	No Global Variables Allowed	No Global Variables
		Only Global Constants	Used
		Meaning Conversions,Physical Constants only	
	11	Static Local	129
	12	Default arguments	1202
	13	Reference Parameters	39
	14	Overloading functions	581
	15	Exit function *******	
7	4	Array Initialization	133
	6	Processing Arrays	1116
	7	Parallel Arrays	28
	8	Arrays as function arguments	28
	9	2-D Arrays	141
	12	STL Vector	138
8	1	Linear and Binary Search	721
	3	Bubble and Selection Sort	Sorting Not Included
	5	Search/Sorting Vectors ******	
****** Not required to show			