# Game of Blackjack

Cody Pfrunder
CSC 5
Spring 2021
43358

# Table of contents

## Introduction

---------------------------------------------------------------------------

*"It's not about the cards you are dealt, but how you play the hand."*

**Blackjack is played against the dealer, who also has a hand of his own. The dealer opens the game by dealing two cards to everyone. A player examines his cards to decide if he wants to hit, split, stand or double down. Though there are several players at a table, most of the time 1 - 4 players, each one is playing the dealer individually. A player may get an undeniably winning hand close to 21 where the player may just sit back and enjoy what is happening in the game with the other players. This game is awesome because it includes skill, good choices, and a little bit of luck. Blackjack is one of the most popular games at a casino and can be a good money maker depending on the amount you are willing to risk.**

## How the game works

---------------------------------------------------------------------------

**Equipment**

Blackjack is played with a standard international deck of cards with the Jokers removed, leaving 52 cards. Originally the game was played with a single deck. Blackjack is now usually offered in either single deck, double deck, 4 deck, 6 deck or 8 deck variants. Besides the cards, the game requires a table, chips, a discard tray, cut card and a shoe.

**Card Values**

When playing Blackjack the numeral cards 2 to 10 have their face values. Jacks, Queens and Kings are valued at 10, and Aces can have a value of either 1 or 11. The Ace is always valued at 11 unless that would result in the hand going over 21, in which case it is valued as 1.
A Blackjack is valued at 21 which would be an ace along with a king, jack, queen, or ten.

**Order of play and playing options**

The player to the left goes first and must decide whether to "stand" (not ask for another card) or "hit" (ask for another card in an attempt to get closer to a count of 21, or even hit 21 exactly). Thus, a player may stand on the two cards originally dealt to them, or they may ask the dealer for additional cards, one at a time, until deciding to stand on the total (if it is 21 or under), or goes "bust" (if it is over 21). In this case the player loses and the dealer collects the bet wagered. The dealer then turns to the next player to their left and serves them in the same manner.

**Payouts**

-Player gets a blackjack and beats the dealer's cards. Then the ratio of the payout is 3:2. For example, if you bet $10 your payout would be $15.

-The payout would be 1:1 if you beat the dealer's hand without a blackjack. For example, you bet $10 you would get another $10

-No money will exchange hands if you and the dealer will get the same value and it is called a push.

-When the dealer gets a better hand than you or a blackjack the dealer will take your bet and you will have to place another wager.

## My Approach to the Game

------------------------------------------------------------------------------------------

**Translating Game Rules to Programming Language**

How do I assign each player a card from a deck of cards?
How will each player know what card they got?
How do I continuously add a new card to the hand sum for each player AND the dealer?
How will a player get blackjack?
How can the bet increase or be rewarded?
How do I allow a player to stand or hit as each turn goes around the table?

These were my very first concerns that arose when deciding on my program logic for blackjack to work properly. The first project without using arrays and functions was very long and strenuous as code had to align play by play so the game could be long enough, fun, and close to the real thing. So, learning and understanding arrays and functions allowed me to shorten my code and follow an order of game play at ease. This also allowed me to create a game including 4 players rather than just one.

**The Similarities from my Version to the Real Thing**

There are several similar components to a real game of blackjack if you were to play in a casino compared to my program/version.
My BLACKJACK program follows these same rules as a normal game:
-The players decide to play, and are prompted to bet.
-Each player is granted two cards left to right at the beginning.
-The dealer's first card is revealed which then allows each player to decide for another card or stand with the two they were initially given.
-The game allows you to hit for a fourth card if the first decision was not to stand.
-Every time a player hits for another the dealer grabs another card.
-Payouts are 3:2 for BLACKJACK and 1:1 for a win.

**Differences from the Real Thing**

One of the biggest differences in my version of Blackjack is that there are no aces, deciding if you or the dealer have a so-called "soft hand". This also makes it more difficult to get a blackjack because your cards will only be valued 1-10, hopefully adding up to 21 at times. These cards valued 1-10 can also be pulled multiple times from different players so you can say my game is not a traditional single deck. The cards also are not assigned clubs, spades, or hearts as I did not see a real meaning when compared to importance of the value. I also did not incorporate doubling down, or splitting a hand. Finally, there is no house advantage. These concepts seemed difficult to add on to my code as it was already so much and set in a good place. I am sure with more guidance and understanding, I could add these aspects of the game very easily.

## The Logic of My Project

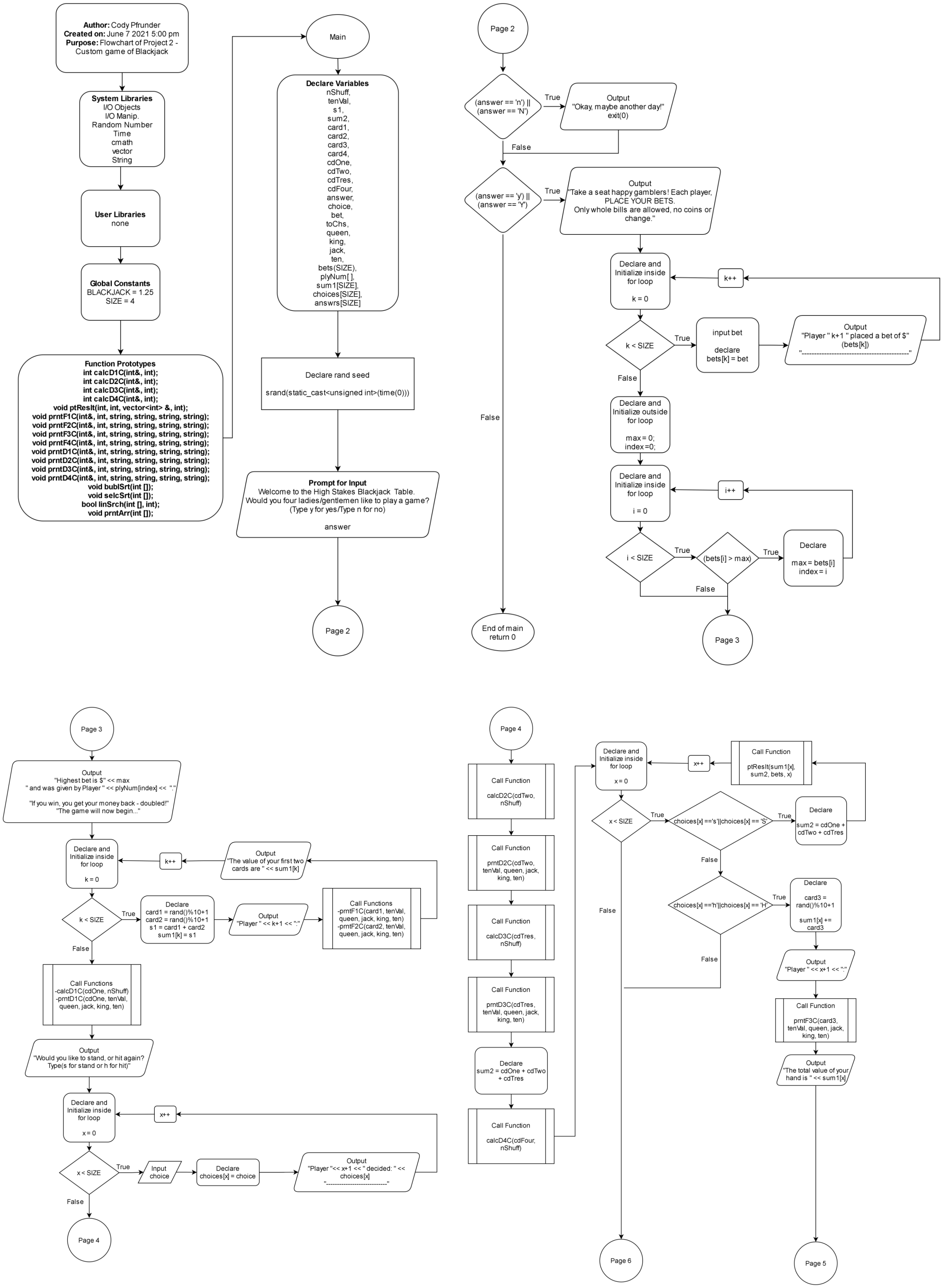
--------------------------------------------------------------------------------

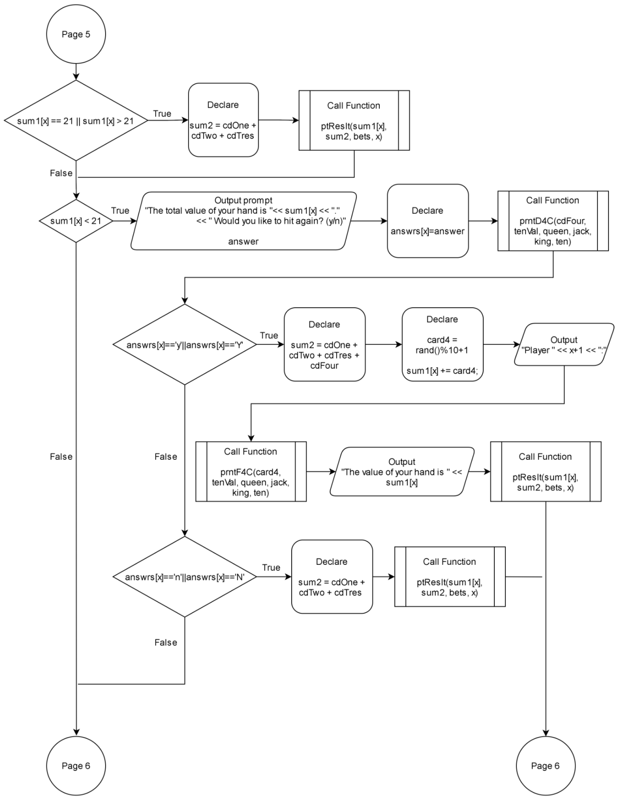
**pseudocode:**

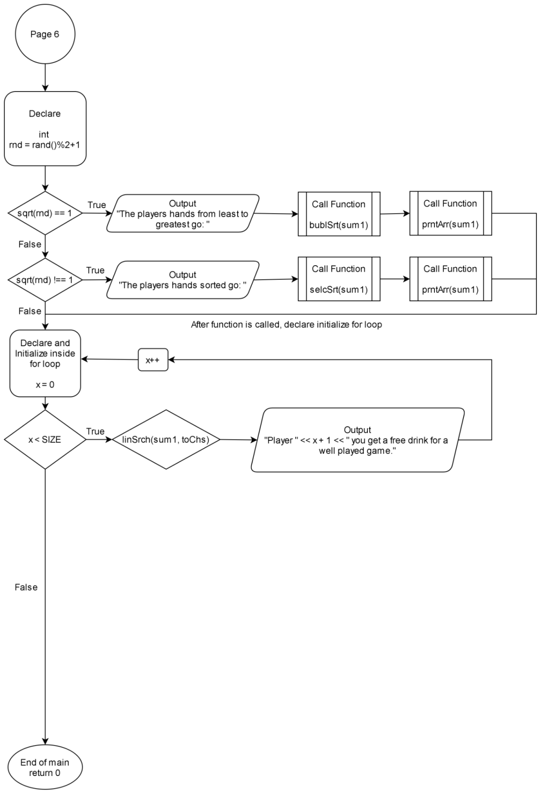
```
Put in opening comments

Bring in system libraries, func.
Prototypes, named constants

Enter main, immediately set random
Seed

Prompt for input to play or not

Player doesn't want to play, exit

Player wants to play, continue

Validate input

4 players input bet

Compare who has the highest bet, output
Player and highest bet

Assign 2 cards to each player's hand

Print the name and value of each card

Calculate and print dealer's first card

Prompt players to hit or stand
```

```
Receive input for each player

Calculate and print dealer’s second card

Calculate dealer’s third card

Calculate dealers fourth card

Loop through each player’s decision to hit
Or stand

When stand, add cards to their sum and call
Function ptReslt( ) to see win or loss

When hit, calc. Third card, add to sum

If it is a blackjack or bust call function
ptReslt( ) to see win or loss

If it is less than 21 ask player if they’d like a
Fourth card

Print dealer’s fourth card

If hit for a fourth card, assign dealer’s fourth card to
His hand

Calc player’s fourth card and add to sum

Print players fourth card and total hand
Call function ptReslt( ) to see win/loss

If answer was no to a fourth card,
Add dealers third card only to hand
 call function ptReslt( ) to see win or loss
```

# Flowchart:


Author: Cody Pfrunder
Created on: June 7 2021 5:00 pm
Purpose: Flowchart of Project 2 - Custom game of Blackjack
System Libraries
I/O Objects
I/O Manip.
Random Number
Time
cmath
vector
String
User Libraries
none
Global Constants
BLACKJACK = 1.25
SIZE = 4
Function Prototypes
int calcD1C(int&, int);
int calcD2C(int&, int);
int calcD3C(int&, int);
int calcD4C(int&, int);
void ptResIt(int, int, vector<int> &, int);
void prntF1C(int&, int, string, string, string, string);
void prntF2C(int&, int, string, string, string, string);
void prntF3C(int&, int, string, string, string, string);
void prntF4C(int&, int, string, string, string, string);
void prntD1C(int&, int, string, string, string, string);
void prntD2C(int&, int, string, string, string, string);
void prntD3C(int&, int, string, string, string, string);
void prntD4C(int&, int, string, string, string, string);
void bublSrt(int []);
void selcSrt(int []);
bool linSrch(int [], int);
void prntArr(int []);
Main
Declare Variables
nShuff,
tenVal,
s1,
sum2,
card1,
card2,
card3,
card4,
cdOne,
cdTwo,
cdTres,
cdFour,
answer,
choice,
bet,
toChs,
queen,
king,
jack,
ten,
bets(SIZE),
plyNum[ ],
sum1[SIZE],
choices[SIZE],
answrs[SIZE]
Declare rand seed
srand(static_cast<unsigned int>(time(0)))
Prompt for Input
Welcome to the High Stakes Blackjack Table.
Would you four ladies/gentlemen like to play a game?
(Type y for yes/Type n for no)
answer
Page 2
Page 2
(answer == 'n') || (answer == 'N')
True
Output
"Okay, maybe another day!"
exit(0)
False
(answer == 'y) || (answer == 'Y')
True
Output
"Take a seat happy gamblers! Each player, PLACE YOUR BETS.
Only whole bills are allowed, no coins or change."
Declare and Initialize inside for loop
k = 0
k++
k < SIZE
True
input bet
declare
bets[k] = bet
Output
"Player " k+1 " placed a bet of $" (bets[k])
"-----------------------------------------"
False
Declare and Initialize outside for loop
max = 0;
index =0;
Declare and Initialize inside for loop
i = 0
i++
i < SIZE
True
(bets[i] > max)
True
Declare
max = bets[i]
index = i
False
False
End of main
return 0
Page 3
Page 3
Output
"Highest bet is $" << max
" and was given by Player " << plyNum[index] << "."
"If you win, you get your money back - doubled!"
"The game will now begin..."
Declare and Initialize inside for loop
k = 0
k++
Output
"The value of your first two cards are " << sum1[k]
k < SIZE
True
Declare
card1 = rand()%10+1
card2 = rand()%10+1
s1 = card1 + card2
sum1[k] = s1
Output
"Player " << k+1 << ":"
Call Functions
-prntF1C(card1, tenVal, queen, jack, king, ten)
-prntF2C(card2, tenVal, queen, jack, king, ten)
False
Call Functions
-calcD1C(cdOne, nShuff)
-prntD1C(cdOne, tenVal, queen, jack, king, ten)
Output
"Would you like to stand, or hit again? Type(s for stand or h for hit)"
Declare and Initialize inside for loop
x = 0
x++
x < SIZE
True
Input choice
Declare
choices[x] = choice
Output
"Player "<< x+1 << " decided: " << choices[x]
"---------------------------"
False
Page 4
Page 4
Call Function
calcD2C(cdTwo, nShuff)
Call Function
prntD2C(cdTwo, tenVal, queen, jack, king, ten)
Call Function
calcD3C(cdTres, nShuff)
Call Function
prntD3C(cdTres, tenVal, queen, jack, king, ten)
Declare
sum2 = cdOne + cdTwo + cdTres
Call Function
calcD4C(cdFour, nShuff)
Declare and Initialize inside for loop
x = 0
x++
Call Function
ptResIt(sum1[x], sum2, bets, x)
x < SIZE
True
choices[x] =='s'||choices[x] == 'S'
True
Declare
sum2 = cdOne + cdTwo + cdTres
False
choices[x] =='h'||choices[x] == 'H'
True
Declare
card3 = rand()%10+1
sum1[x] += card3
False
False
Output
"Player " << x+1 << ":"
Call Function
prntF3C(card3, tenVal, queen, jack, king, ten)
Output
"The total value of your hand is " << sum1[x]
Page 6
Page 5


Page 5
sum1[x] == 21 || sum1[x] > 21
True
Declare
sum2 = cdOne + cdTwo + cdTres
Call Function
ptReslt(sum1[x], sum2, bets, x)
False
sum1[x] < 21
True
Output prompt
"The total value of your hand is "<< sum1[x] << "." << " Would you like to hit again? (y/n)"
answer
Declare
answrs[x]=answer
Call Function
prntD4C(cdFour, tenVal, queen, jack, king, ten)
answrs[x]=='y'||answrs[x]=='Y'
True
Declare
sum2 = cdOne + cdTwo + cdTres + cdFour
Declare
card4 = rand()%10+1
sum1[x] += card4;
Output
"Player " << x+1 << ":"
False
False
Call Function
prntF4C(card4, tenVal, queen, jack, king, ten)
Output
"The value of your hand is " << sum1[x]
Call Function
ptReslt(sum1[x], sum2, bets, x)
answrs[x]=='n'||answrs[x]=='N'
True
Declare
sum2 = cdOne + cdTwo + cdTres
Call Function
ptReslt(sum1[x], sum2, bets, x)
False
Page 6
Page 6


Page 6
Declare
int
rnd = rand()%2+1
sqrt(rnd) == 1
True
Output
"The players hands from least to greatest go: "
Call Function
bublSrt(sum1)
Call Function
prntArr(sum1)
False
sqrt(rnd) !== 1
True
Output
"The players hands sorted go: "
Call Function
selcSrt(sum1)
Call Function
prntArr(sum1)
False
After function is called, declare initialize for loop
Declare and Initialize inside for loop
x = 0
x++
x < SIZE
True
linSrch(sum1, toChs)
Output
"Player " << x + 1 << " you get a free drink for a well played game."
False
End of main
return 0

## Constructs and Concepts Utilized / Checklist

# Cross Reference Project 1

### AUTHOR : CODY PFRUNDER * CSC5 PROF. LEHR BLACKJACK PROJECT 2

| Chapter | Section | Topic | Where Line #"s | Pts | Notes |
|---|---|---|---|---|---|
| 2 | 2 | cout | 91 | | |
| | 3 | libraries | 9-15 | 5 | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
| | 4 | variables/literals | 52-73 | | No variables in global area, failed project! |
| | 5 | Identifiers | | | |
| | 6 | Integers | 55-68 | 1 | |
| | 7 | Characters | 52, 53 | 1 | |
| | 8 | Strings | 70-73 | 1 | |
| | 9 | Floats No Doubles | 22 | 1 | Using doubles will fail the project, floats OK! |
| | 10 | Bools | 602 | 1 | |
| | 11 | Sizeof ***** | | | |
| | 12 | Variables 7 characters or less | | | All variables <= 7 characters |
| | 13 | Scope ***** No Global Variables | | | |
| | 14 | Arithmetic operators | | | |
| | 15 | Comments 20%+ | Throughout code | 2 | Model as pseudo code |
| | 16 | Named Constants | 22, 23 | | All Local, only Conversions/Physics/Math in Global area |
| | 17 | Programming Style ***** Emulate | | | Emulate style in book/in class repositiory |
| | | | | | |
| 3 | 1 | cin | 94 | | |
| | 2 | Math Expression | 271 | | |
| | 3 | Mixing data types **** | | | |
| | 4 | Overflow/Underflow **** | | | |
| | 5 | Type Casting | 111 | 1 | |
| | 6 | Multiple assignment ***** | | | |
| | 7 | Formatting output | 109 | 1 | |

| | | | | | |
|---|---|---|---|---|---|
| | 8 | Strings | 70-73<br>347 | 1 | |
| | 9 | Math Library | 13 | 1 | All libraries included have to be used |
| | 10 | Hand tracing ****** | | | |
| | | | | | |
| 4 | 1 | Relational Operators | | | |
| | 2 | if | 98 | 1 | Independent if |
| | 4 | If-else | 271-278 | 1 | |
| | 5 | Nesting | All throughout code | 1 | |
| | 6 | If-else-if | 199-207 | 1 | |
| | 7 | Flags ***** | | | |
| | 8 | Logical operators | 98 | 1 | |
| | 11 | Validating user input | 98-100 | 1 | |
| | 13 | Conditional Operator | Throughout code | 1 | |
| | 14 | Switch | 346 | 1 | |
| | | | | | |
| 5 | 1 | Increment/Decrement | Throughout code | 1 | |
| | 2 | While | 192 | 1 | |
| | 5 | Do-while | 603-613 | 1 | |
| | 6 | For loop | Throughout code | 1 | |
| | 11 | Files input/output both | Not sure | 2 | |
| | 12 | No breaks in loops ****** | | | Failed Project if included |
| | | | | | |
| | | | | | |
| | Not ed to | | Total | 30 | |

# Cross Reference for Project 2

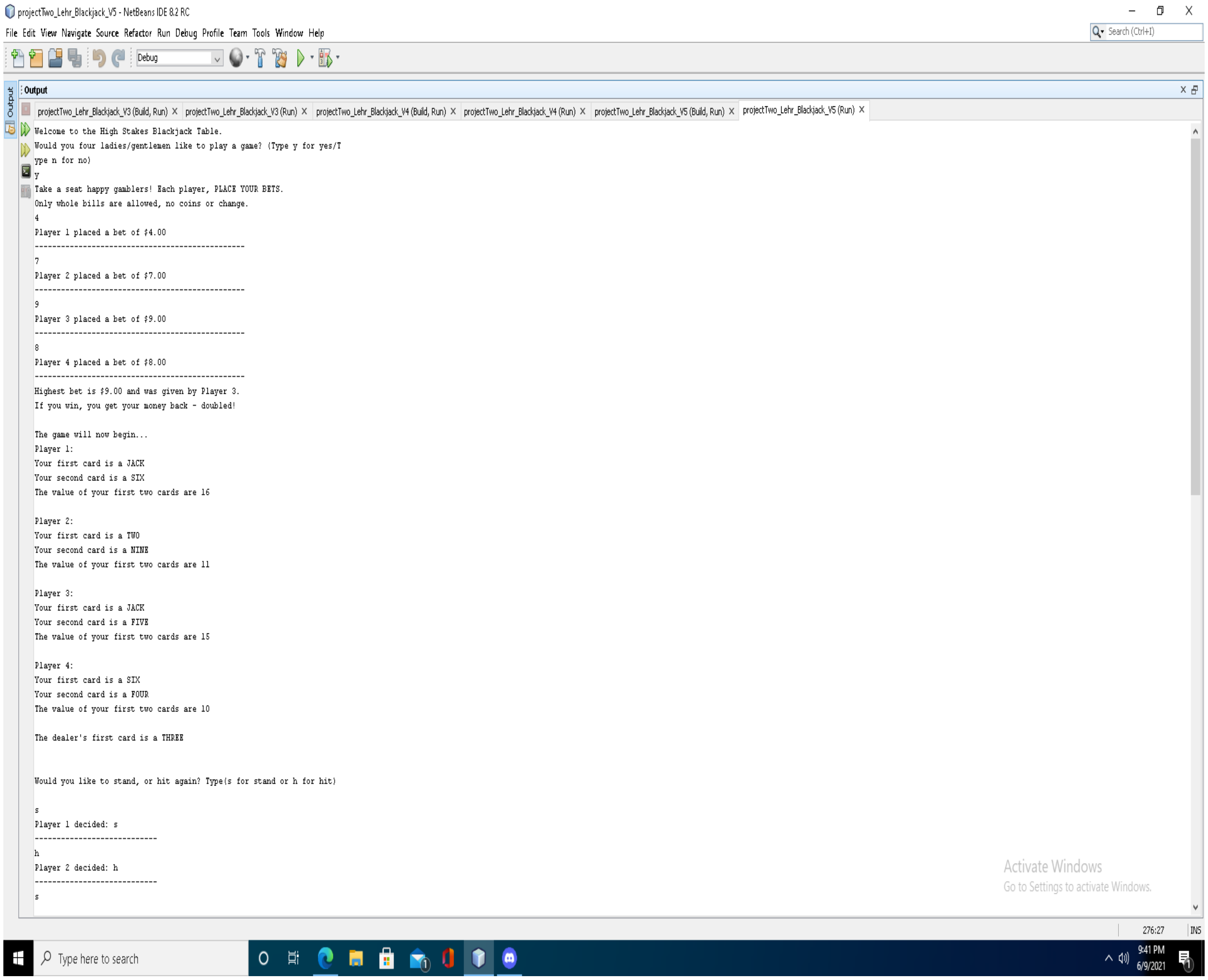
## You are to fill-in with where located in code

| Chapter | Section | Topic | Where Line #"s | Pts | Notes |
|---|---|---|---|---|---|
| 6 | | Functions | | | |
| | 3 | Function Prototypes | 26 | 4 | Always use prototypes |
| | 5 | Pass by Value | Throughout code | 4 | |
| | 8 | return | 302 | 4 | A value from a function |
| | 9 | returning boolean | 634 | 4 | |
| | 10 | Global Variables | | XXX | Do not use global variables -100 pts |
| | 11 | static variables | 115, 116 | 4 | |
| | 12 | defaulted arguments | Throughout functions | 4 | |
| | 13 | pass by reference | 298, throughout | 4 | |
| | 14 | overloading | 125 | 5 | |
| | 15 | exit() function | 100 | 4 | |
| 7 | | Arrays | | | |
| | 1 to 6 | Single Dimensioned Arrays | 77-80 used throughout | 3 | |
| | 7 | Parallel Arrays | 76, 126 | 2 | |
| | 8 | Single Dimensioned as Function Arguments | 601 | 2 | |
| | 9 | 2 Dimensioned Arrays | | 2 | Emulate style in book/in class repositiory |
| | 12 | STL Vectors | 75, 535 | 2 | |
| | | Passing Arrays to and from Functions | 535, 601, several others | 5 | |
| | | Passing Vectors to and from Functions | 75, 535 | 5 | |
| | | | | | |
| 8 | | Searching and Sorting Arrays | | | |
| | 3 | Bubble Sort | 601, 274 | 4 | |
| | 3 | Selection Sort | 616, 281 | 4 | |
| | 1 | Linear or Binary Search | 630, 287 | 4 | |
| | | | | | |

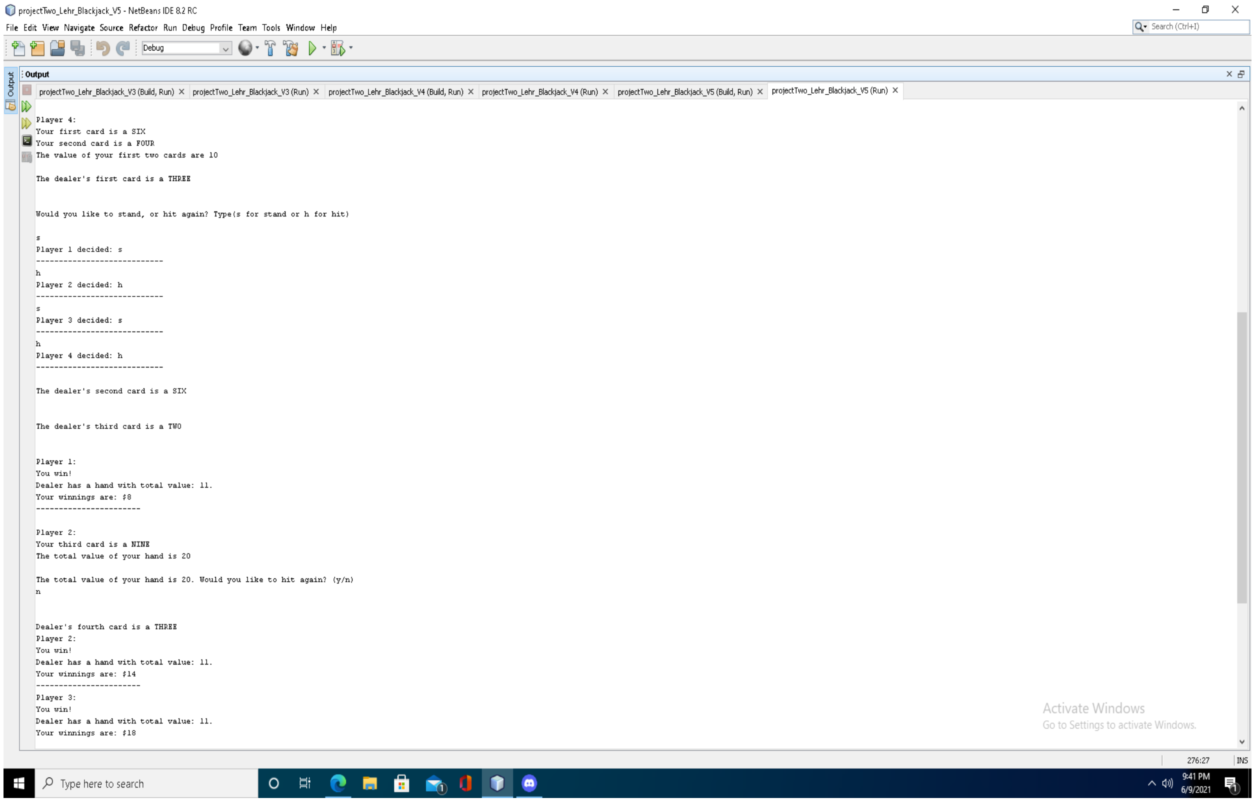
| | | | | | |
|---|---|---|---|---|---|
| | ot req. | | Total | 70 | Other 30 points from Proj 1 first sheet tab |

## Proof of a working product

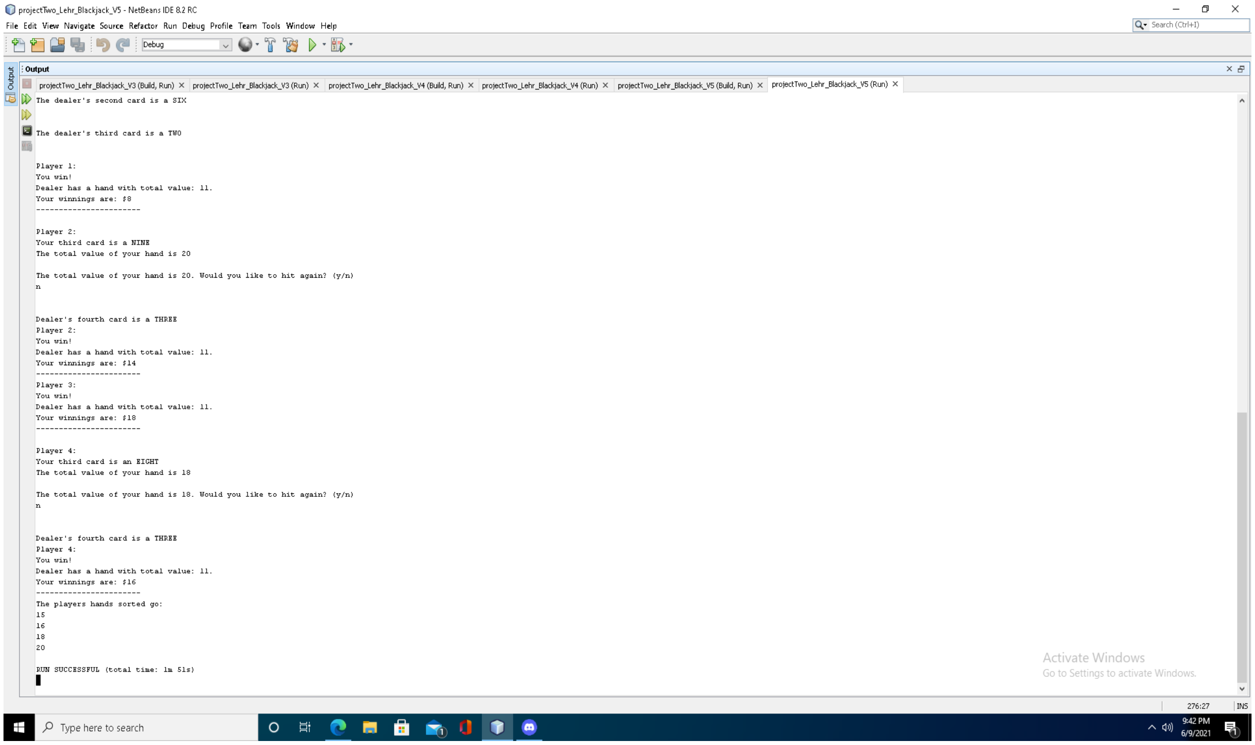
In the event, that my program does not work once it reaches Dr. Lehr, I have provided some screenshots that prove that the program did work at one time on the next few pages.


projectTwo_Lehr_Blackjack_V5 - NetBeans IDE 8.2 RC
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Debug
Output
projectTwo_Lehr_Blackjack_V3 (Build, Run)
projectTwo_Lehr_Blackjack_V3 (Run)
projectTwo_Lehr_Blackjack_V4 (Build, Run)
projectTwo_Lehr_Blackjack_V4 (Run)
projectTwo_Lehr_Blackjack_V5 (Build, Run)
projectTwo_Lehr_Blackjack_V5 (Run)
Activate Windows
Go to Settings to activate Windows.
276:27
INS
Type here to search
9:41 PM
6/9/2021

```
Welcome to the High Stakes Blackjack Table.
Would you four ladies/gentlemen like to play a game? (Type y for yes/T
ype n for no)
y
Take a seat happy gamblers! Each player, PLACE YOUR BETS.
Only whole bills are allowed, no coins or change.
4
Player 1 placed a bet of $4.00
------------------------------------------------
7
Player 2 placed a bet of $7.00
------------------------------------------------
9
Player 3 placed a bet of $9.00
------------------------------------------------
8
Player 4 placed a bet of $8.00
------------------------------------------------
Highest bet is $9.00 and was given by Player 3.
If you win, you get your money back - doubled!

The game will now begin...
Player 1:
Your first card is a JACK
Your second card is a SIX
The value of your first two cards are 16

Player 2:
Your first card is a TWO
Your second card is a NINE
The value of your first two cards are 11

Player 3:
Your first card is a JACK
Your second card is a FIVE
The value of your first two cards are 15

Player 4:
Your first card is a SIX
Your second card is a FOUR
The value of your first two cards are 10

The dealer's first card is a THREE


Would you like to stand, or hit again? Type(s for stand or h for hit)

s
Player 1 decided: s
----------------------------
h
Player 2 decided: h
----------------------------
s
```


projectTwo_Lehr_Blackjack_V5 - NetBeans IDE 8.2 RC
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Debug
Output
projectTwo_Lehr_Blackjack_V3 (Build, Run)
projectTwo_Lehr_Blackjack_V3 (Run)
projectTwo_Lehr_Blackjack_V4 (Build, Run)
projectTwo_Lehr_Blackjack_V4 (Run)
projectTwo_Lehr_Blackjack_V5 (Build, Run)
projectTwo_Lehr_Blackjack_V5 (Run)
Activate Windows
Go to Settings to activate Windows.
276:27
INS
Type here to search
9:41 PM
6/9/2021

```
Player 4:
Your first card is a SIX
Your second card is a FOUR
The value of your first two cards are 10

The dealer's first card is a THREE


Would you like to stand, or hit again? Type(s for stand or h for hit)

s
Player 1 decided: s
----------------------------
h
Player 2 decided: h
----------------------------
s
Player 3 decided: s
----------------------------
h
Player 4 decided: h
----------------------------

The dealer's second card is a SIX


The dealer's third card is a TWO


Player 1:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $8
-----------------------

Player 2:
Your third card is a NINE
The total value of your hand is 20

The total value of your hand is 20. Would you like to hit again? (y/n)
n


Dealer's fourth card is a THREE
Player 2:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $14
-----------------------
Player 3:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $18
```


projectTwo_Lehr_Blackjack_V5 - NetBeans IDE 8.2 RC
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Debug
Output
projectTwo_Lehr_Blackjack_V3 (Build, Run)
projectTwo_Lehr_Blackjack_V3 (Run)
projectTwo_Lehr_Blackjack_V4 (Build, Run)
projectTwo_Lehr_Blackjack_V4 (Run)
projectTwo_Lehr_Blackjack_V5 (Build, Run)
projectTwo_Lehr_Blackjack_V5 (Run)
Activate Windows
Go to Settings to activate Windows.
276:27
INS
Type here to search
9:42 PM
6/9/2021

```
The dealer's second card is a SIX


The dealer's third card is a TWO


Player 1:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $8
-----------------------

Player 2:
Your third card is a NINE
The total value of your hand is 20

The total value of your hand is 20. Would you like to hit again? (y/n)
n


Dealer's fourth card is a THREE
Player 2:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $14
-----------------------
Player 3:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $18
-----------------------

Player 4:
Your third card is an EIGHT
The total value of your hand is 18

The total value of your hand is 18. Would you like to hit again? (y/n)
n


Dealer's fourth card is a THREE
Player 4:
You win!
Dealer has a hand with total value: 11.
Your winnings are: $16
-----------------------
The players hands sorted go:
15
16
18
20

RUN SUCCESSFUL (total time: 1m 51s)
```

## References

1. Dr. Lehr's Lectures & Lab
2. “Starting Out with C++: From Control Structures through Objects” Gaddis, Tony. 8th Edition. (Textbook)
3. “Problem Solving with C++” Savitch, Walter. 10th Edition. (Textbook)

## Program

```
/*
    Author:    Cody Pfrunder
    Date:      7 June 2021  10:35 am
    Project 2: Blackjack Game
    Version:   v.5 (Final Version)
 */

//System Libraries
#include <iostream>    //Input/Output Library
#include <iomanip>     //Input/Output Manipulation
#include <cstdlib>     //Random Number Library
#include <ctime>       //Time Library
#include <cmath>       //cmath library
#include <string>      //String Library
#include <vector>      //Vector Library
using namespace std;   //Library Name-space

//User Libraries

//Global/Universal Constants -- No Global Variables
//Science, Math, Conversions, Higher Dimensioned constants only
const float BLACKJACK = 1.25; //Pay increase for a blackjack
const int   SIZE = 4;         //Amt of players

//Function Prototypes
int calcD1C(int&, int);
int calcD2C(int&, int);
int calcD3C(int&, int);
int calcD4C(int&, int);
void ptReslt(int, int, vector<int> &, int);
void prntF1C(int&, int, string, string, string, string);
void prntF2C(int&, int, string, string, string, string);
void prntF3C(int&, int, string, string, string, string);
void prntF4C(int&, int, string, string, string, string);
void prntD1C(int&, int, string, string, string, string);
void prntD2C(int&, int, string, string, string, string);
void prntD3C(int&, int, string, string, string, string);
void prntD4C(int&, int, string, string, string, string);
void bublSrt(int []);
void selcSrt(int []);
bool linSrch(int [], int);
void prntArr(int []);


//Execution Begins Here
int main(int argc, char** argv){
```

```
//Set the Random number seed
srand(static_cast<unsigned int>(time(0)));

//Declare variables
char   answer,   //Answer to play or not
       choice;   //Choice to stand with hand, or hit for more cards

int    nShuff,   //Number of card shuffles
       bet,      //Player's bet
       tenVal,   //Cards with value of ten
       s1,       //Sum of card values (Player's hand)
       sum2,     //Sum of card values (Dealer's hand)
       card1,    //Player's first card
       card2,    //Player's second card
       card3,    //Player's third card if decided to hit.
       card4,    //Player's fourth card if decided to hit twice
       cdOne,    //Dealer's first card
       cdTwo,    //Dealer's second card
       cdTres,   //Dealer's third card if value is less than 21
       cdFour,   //Dealer's fourth card if value is less than 21
       toChs;    //Player chosen with BJ at end of the game for a drink

string king,     //king face card
       queen,    //queen face card
       jack,     //jack face card
       ten;      //ten card

vector<int> bets(SIZE);           //4 players bets array
int    plyNum[ ] = {1 , 2, 3, 4}; //Parallel array to bets[] - highest bet
int    sum1[SIZE];                //4 players card value sums

char   choices[SIZE]; //4 players choice to stand or hit array
char   answrs[SIZE];  //4 Players answer to hit for fourth card after third

//Initialize variables
toChs = 21;            //Player with blackjack gets a sticker
nShuff = 1;            //1 random card shuffle
king = "KING";         //King card
queen = "QUEEN";       //Queen card
jack = "JACK";         //Jack card
ten = "TEN";           //Ten card


cout << "Welcome to the High Stakes Blackjack Table.\n";
cout << "Would you four ladies/gentlemen like to play a game? "
     << "(Type y for yes/Type n for no)\n";
cin >> answer;

//Process, map inputs to outputs
//Answers no
if((answer == 'n') || (answer == 'N')){
    cout<<"Okay, maybe another day!\n";
    exit(1);
}
```

```
//Answers yes
if((answer == 'y') || (answer == 'Y')){
    cout << "Take a seat happy gamblers! Each player, PLACE YOUR BETS.\n";
    cout << "Only whole bills are allowed, no coins or change.\n";
    for (int k = 0; k < SIZE; k++){
        cin>>bet;
        bets[k]=bet;
        cout << fixed << setprecision(2) << showpoint;
        cout << "Player "<<k+1<<" placed a bet of $"
             << static_cast<float>(bets[k])<<endl;
        cout << "------------------------------------------------\n";
    }

    static int max = 0;
    static int index = 0;
    for(int i = 0; i < SIZE; i++){
        if (bets[i] > max){
            max = bets[i];
            index = i;
        }
    }

    cout << fixed << setprecision(2) << showpoint;
    cout << "Highest bet is $"<< static_cast<float>(max)
         << " and was given by Player " << plyNum[index] << ".\n";

    cout << "If you win, you get your money back - doubled!\n\n";
    cout << "The game will now begin...\n";

    //4 Players get first two cards
    for(int k = 0; k < SIZE; k++){
        //Calculate first card for player
        card1 = rand()%10+1;  //[1, 10]

        //Calculate second card for player
        card2 = rand()%10+1;  //[1, 10]

        //Begin to store sum
        s1 = card1 + card2;

        //Assign a sum to 4 players
        sum1[k] = s1;

        //Align player's first and second card to face number/name
        cout << "Player " << k+1 << ":\n";
        prntF1C(card1, tenVal, queen, jack, king, ten);
        prntF2C(card2, tenVal, queen, jack, king, ten);
        cout << "The value of your first two cards are " << sum1[k] << endl
             << endl;
    }

    //Call calcD1C func. for dealer's first card
    calcD1C(cdOne, nShuff);

    //Align & print dealer's first card to name using prntD2C func.
```

```
prntD1C(cdOne, tenVal, queen, jack, king, ten);
cout << endl << endl;

//Prompt player to stand with current hand or get a third card
cout << "Would you like to stand, or hit again? Type(s for stand "
     <<   "or h for hit)\n\n";

//Assign choice (s for stand/h for hit) to each player
for(int x = 0; x < SIZE; x++){
    cin>>choice;
    choices[x]=choice;
    cout << "Player "<< x+1 << " decided: " << choices[x] << endl;
    cout << "----------------------------\n";
}
cout << endl;

//Call calcD2C func. for dealer's second card reveal
calcD2C(cdTwo, nShuff);

//Align & print dealer's second card name/number
prntD2C(cdTwo, tenVal, queen, jack, king, ten);
cout << endl << endl;

//Call calcD3C func. for dealer's third card
calcD3C(cdTres, nShuff);

//Print dealer's third card and add to hand
int n = 1;
do{
    prntD3C(cdTres, tenVal, queen, jack, king, ten);
    cout << endl << endl;

    sum2 = cdOne + cdTwo + cdTres;

    n++;
}while(n<=nShuff);

//Calc dealer's fourth card
calcD4C(cdFour, nShuff);

//Player chose to stand
for(int x = 0; x < SIZE; x++){
    if(choices[x] == 's'||choices[x] == 'S'){
        sum2 = cdOne + cdTwo + cdTres;
        //Who wins? Func. ptResult()
        ptReslt(sum1[x], sum2, bets, x);
        cout << "-----------------------" << endl
             << endl;
    }

    else if(choices[x] == 'h'||choices[x] == 'H'){
        //Calculate third card for player
        card3 = rand()%10+1;  //[1, 10]

        //Assign sum to player that hit
```

```
sum1[x] += card3;

//Align player's third card to face number/name
cout << "Player " << x+1 << ":\n";
prntF3C(card3, tenVal, queen, jack, king, ten);
cout << "The total value of your hand is " << sum1[x] << endl
     << endl;

if(sum1[x] == 21 || sum1[x] > 21){
    sum2 = cdOne + cdTwo + cdTres;
    //Who wins? Func. ptResult()
    ptReslt(sum1[x], sum2, bets, x);
    cout << "-----------------------" << endl;
}

//Not a blackjack
else if(sum1[x] < 21){
    cout << "The total value of your hand is "<< sum1[x] << "."
         << " Would you like to hit again? (y/n)\n";

    cin >> answer;
    answrs[x]=answer;
    cout << endl << endl;

    //Name dealer's card type
    prntD4C(cdFour, tenVal, queen, jack, king, ten);

    //Player chooses to hit for a fourth card
    if (answrs[x]=='y'||answrs[x]=='Y'){
        //Add dealer's fourth card to hand
        sum2 = cdOne + cdTwo + cdTres + cdFour;

        //Calculate fourth card for player
        card4 = rand()%10+1;  //[1, 10]
        //Assign sum to player that hit
        sum1[x] += card4;

        //Align player's fourth card to face number/name
        cout << "Player " << x+1 << ":\n";
        prntF4C(card4, tenVal, queen, jack, king, ten);
        cout << "The value of your hand is " << sum1[x] << endl
             << endl;

        //Who wins? Func. ptResult()
        ptReslt(sum1[x], sum2, bets, x);
        cout << "-----------------------" << endl;
    }
    //Player chooses not to hit for a fourth card
    else if (answrs[x]=='n'||answrs[x]=='N'){
        //Add dealers third card to hand
        sum2 = cdOne + cdTwo + cdTres;
        ptReslt(sum1 [x], sum2, bets, x);
        cout << "-----------------------" << endl;
    }
}
```

```
            }
        }

        int rnd = rand()%2+1; //Random number 1 or 2
        if(sqrt(rnd) == 1){
            cout << "The players hands from least to greatest go: \n";
            //Bubble sort players hands least to greatest
            bublSrt(sum1);
            // Function to print an array
            prntArr(sum1);
        }
        else{
            cout << "The players hands sorted go: \n";
            //Selection sort players hands least to greatest
            selcSrt(sum1);
            // Function to print an array
            prntArr(sum1);
        }
        //Linear search to get a free drink
        for(int x = 0; x < SIZE; x++){
            if(linSrch(sum1, toChs)){
                cout << "Player " << x + 1 << " you get a free drink for a "
                     << "well played game." << endl;
            }
        }
    }
    //Exit stage right
    return 0;
}

//Dealer's hand (first card)
int calcD1C(int &cdOne, int nShuff){
    for(int n=1; n<=nShuff; n++){
        cdOne = rand()%10+1; //[1, 10]
    }
    return cdOne;
}

//Dealer's hand (second card)
int calcD2C(int &cdTwo, int nShuff){
    for(int n=1; n<=nShuff; n++){
        cdTwo = rand()%10+1; //[1, 10]
    }
    return cdTwo;
}

//Dealer's hand (third card)
int calcD3C(int &cdTres, int nShuff){
    for(int n=1; n<=nShuff; n++){
        cdTres = rand()%10+1; //[1, 10]
    }
    return cdTres;
}

//Dealer's hand (fourth card)
```

```
int calcD4C(int &cdFour, int nShuff){
    for(int n=1; n<=nShuff; n++){
        cdFour = rand()%10+1; //[1, 10]
    }
    return cdFour;
}

//Print player's first card
void prntF1C(int &card1, int tenVal, string queen, string jack, string king,
        string ten){
    //Name card type
    switch(card1){
        case 1: cout<<"Your first card is a ONE\n";break;
        case 2: cout<<"Your first card is a TWO\n";break;
        case 3: cout<<"Your first card is a THREE\n";break;
        case 4: cout<<"Your first card is a FOUR\n";break;
        case 5: cout<<"Your first card is a FIVE\n";break;
        case 6: cout<<"Your first card is a SIX\n";break;
        case 7: cout<<"Your first card is a SEVEN\n";break;
        case 8: cout<<"Your first card is an EIGHT\n";break;
        case 9: cout<<"Your first card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(card1==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Your first card is a "<<queen<<endl;break;
                    case 2: cout<<"Your first card is a "<<jack<<endl;break;
                    case 3: cout<<"Your first card is a "<<king<<endl;break;
                    case 4: cout<<"Your first card is a "<<ten<<endl;break;
                }
            }break;
    }
}

void prntF2C(int &card2, int tenVal, string queen, string jack, string king,
        string ten){
    switch(card2){
        case 1: cout<<"Your second card is a ONE\n";break;
        case 2: cout<<"Your second card is a TWO\n";break;
        case 3: cout<<"Your second card is a THREE\n";break;
        case 4: cout<<"Your second card is a FOUR\n";break;
        case 5: cout<<"Your second card is a FIVE\n";break;
        case 6: cout<<"Your second card is a SIX\n";break;
        case 7: cout<<"Your second card is a SEVEN\n";break;
        case 8: cout<<"Your second card is an EIGHT\n";break;
        case 9: cout<<"Your second card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(card2==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Your second card is a "<<queen<<endl;break;
                    case 2: cout<<"Your second card is a "<<jack<<endl;break;
                    case 3: cout<<"Your second card is a "<<king<<endl;break;
                    case 4: cout<<"Your second card is a "<<ten<<endl;break;
                }
            }break;
    }
}

void prntF3C(int &card3, int tenVal, string queen, string jack, string king,
        string ten){
    switch(card3){
        case 1: cout<<"Your third card is a ONE\n";break;
        case 2: cout<<"Your third card is a TWO\n";break;
        case 3: cout<<"Your third card is a THREE\n";break;
        case 4: cout<<"Your third card is a FOUR\n";break;
        case 5: cout<<"Your third card is a FIVE\n";break;
        case 6: cout<<"Your third card is a SIX\n";break;
        case 7: cout<<"Your third card is a SEVEN\n";break;
        case 8: cout<<"Your third card is an EIGHT\n";break;
        case 9: cout<<"Your third card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(card3==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Your third card is a "<<queen<<endl;break;
                    case 2: cout<<"Your third card is a "<<king<<endl;break;
                    case 3: cout<<"Your third card is a "<<jack<<endl;break;
                    case 4: cout<<"Your third card is a "<<ten<<endl;break;
                }
            }break;
    }
}

void prntF4C(int &card4, int tenVal, string queen, string jack, string king,
        string ten){
    //Name card type
    switch(card4){
        case 1: cout<<"Your fourth card is a ONE\n";break;
        case 2: cout<<"Your fourth card is a TWO\n";break;
        case 3: cout<<"Your fourth card is a THREE\n";break;
        case 4: cout<<"Your fourth card is a FOUR\n";break;
        case 5: cout<<"Your fourth card is a FIVE\n";break;
        case 6: cout<<"Your fourth card is a SIX\n";break;
        case 7: cout<<"Your fourth card is a SEVEN\n";break;
        case 8: cout<<"Your fourth card is an EIGHT\n";break;
        case 9: cout<<"Your fourth card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(card4==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Your fourth card is a "<<queen<<endl;break;
                    case 2: cout<<"Your fourth card is a "<<king<<endl;break;
                    case 3: cout<<"Your fourth card is a "<<jack<<endl;break;
                    case 4: cout<<"Your fourth card is a "<<ten<<endl;break;
                }
            }break;
    }
}
```

```
//Print dealer's first card
void prntD1C(int &cdOne, int tenVal, string queen, string jack,
        string king, string ten){
    switch(cdOne){
        case 1: cout<<"The dealer's first card is a ONE\n";break;
        case 2: cout<<"The dealer's first card is a TWO\n";break;
        case 3: cout<<"The dealer's first card is a THREE\n";break;
        case 4: cout<<"The dealer's first card is a FOUR\n";break;
        case 5: cout<<"The dealer's first card is a FIVE\n";break;
        case 6: cout<<"The dealer's first card is a SIX\n";break;
        case 7: cout<<"The dealer's first card is a SEVEN\n";break;
        case 8: cout<<"The dealer's first card is an EIGHT\n";break;
        case 9: cout<<"The dealer's first card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(cdOne==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Dealer's first card is a "<<queen<<endl;break;
                    case 2: cout<<"Dealer's first card is a "<<jack<<endl;break;
                    case 3: cout<<"Dealer's first card is a "<<king<<endl;break;
                    case 4: cout<<"Dealer's first card is a "<<ten<<endl;break;
                }
            }break;
    }
}
//Print dealer's second card
void prntD2C(int &cdTwo, int tenVal, string queen, string jack,
        string king, string ten){
    switch(cdTwo){
        case 1: cout<<"The dealer's second card is a ONE\n";break;
        case 2: cout<<"The dealer's second card is a TWO\n";break;
        case 3: cout<<"The dealer's second card is a THREE\n";break;
        case 4: cout<<"The dealer's second card is a FOUR\n";break;
        case 5: cout<<"The dealer's second card is a FIVE\n";break;
        case 6: cout<<"The dealer's second card is a SIX\n";break;
        case 7: cout<<"The dealer's second card is a SEVEN\n";break;
        case 8: cout<<"The dealer's second card is an EIGHT\n";break;
        case 9: cout<<"The dealer's second card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(cdTwo==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Dealer's second card is a "<<queen<<endl;break;
                    case 2: cout<<"Dealer's second card is a "<<jack<<endl;break;
                    case 3: cout<<"Dealer's second card is a "<<king<<endl;break;
                    case 4: cout<<"Dealer's second card is a "<<ten<<endl;break;
                }
            }break;
    }
}

//Print dealers third card
void prntD3C(int &cdTres, int tenVal, string queen, string jack,
        string king, string ten){
    //Name card type
    switch(cdTres){
        case 1: cout<<"The dealer's third card is a ONE\n";break;
        case 2: cout<<"The dealer's third card is a TWO\n";break;
        case 3: cout<<"The dealer's third card is a THREE\n";break;
        case 4: cout<<"The dealer's third card is a FOUR\n";break;
        case 5: cout<<"The dealer's third card is a FIVE\n";break;
        case 6: cout<<"The dealer's third card is a SIX\n";break;
        case 7: cout<<"The dealer's third card is a SEVEN\n";break;
        case 8: cout<<"The dealer's third card is an EIGHT\n";break;
        case 9: cout<<"The dealer's third card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(cdTres==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<< "Dealer's third card is a "<<queen<<endl;break;
                    case 2: cout<< "Dealer's third card is a "<<king<<endl;break;
                    case 3: cout<< "Dealer's third card is a "<<jack<<endl;break;
                    case 4: cout<< "Dealer's third card is a "<<ten<<endl;break;
                }
            }break;
    }
}

void prntD4C(int &cdFour, int tenVal, string queen, string jack, string king,
        string ten){
    switch(cdFour){
        case 1: cout<<"Dealer's fourth card is a ONE\n";break;
        case 2: cout<<"Dealer's fourth card is a TWO\n";break;
        case 3: cout<<"Dealer's fourth card is a THREE\n";break;
        case 4: cout<<"Dealer's fourth card is a FOUR\n";break;
        case 5: cout<<"Dealer's fourth card is a FIVE\n";break;
        case 6: cout<<"Dealer's fourth card is a SIX\n";break;
        case 7: cout<<"Dealer's fourth card is a SEVEN\n";break;
        case 8: cout<<"Dealer's fourth card is an EIGHT\n";break;
        case 9: cout<<"Dealer's fourth card is a NINE\n";break;
        case 10: //Special case - cards with value 10
            if(cdFour==10){
                tenVal = rand()%4+1;
                switch(tenVal){
                    case 1: cout<<"Dealer's fourth card is a "<<queen<<endl;break;
                    case 2: cout<<"Dealer's fourth card is a "<<king<<endl;break;
                    case 3: cout<<"Dealer's fourth card is a "<<jack<<endl;break;
                    case 4: cout<<"Dealer's fourth card is a "<<ten<<endl;break;
                }
            }break;
    }
}

void ptReslt(int sum1, int sum2, vector<int> &bets, int k){
    cout << "Player " << k + 1 <<": \n";
    //Dealer wins by greater
    if(sum2 > sum1 && sum2 < 21){
        cout << "You lose...\n";
        cout << "Dealer wins with value of " << sum2 <<".\n";
    }
```

```
//Player bust
else if(sum1 > 21){
    cout << "Your hand holds the total value: " << sum1 << ".\n";
    cout << "You lose with a busted hand. Dealer wins.\n";
}
//Dealer bust
else if(sum2 > 21 && sum1 != 21){
    cout << "Dealer has a hand with total value: " << sum2 << ".\n";
    cout << "You win! the dealer busts.\n";
    bets[k] *= 2.0;
    cout << "Your winnings are: $" << bets[k] << endl;
}
//Player and dealer have same hand
else if(sum1 == sum2){
    cout << "Dealer has a hand with total value: "<< sum2 << ".\n"
         << "You and the dealer have the same hand.\n"
         << "You get your money back with no damage done!\n";
}
//Player wins
else if(sum1 > sum2 && sum1 < 21){
    cout << "You win!\n"
         << "Dealer has a hand with total value: "<< sum2 << ".\n"
         << "Your winnings are: $" << (bets[k]*2) << endl;
}
//Player got blackjack
else if(sum1 == 21 && sum1 > sum2){
    cout << "The total value of your hand is " << sum1 << ".\n"
         << "You got BLACKJACK!\n"
         << "Dealer has a hand with total value: " << sum2 << ".\n"
         << "Because you got BLACKJACK your winnings are "
         << "increased by 1.25x.\n";
    bets[k] = bets[k] + (bets[k]*BLACKJACK);
    cout << "Your winnings are: $" << bets[k] << endl;
}
//Player got blackjack and dealer busts
else if(sum1 == 21 && sum2 > 21){
    cout << "The total value of your hand is " << sum1 << ".\n"
         << "You got BLACKJACK!\n"
         << "Dealer has a hand with total value: " << sum2 << ".\n"
         << "Because you got BLACKJACK your winnings are "
         << "increased by 1.25x.\n";
    bets[k] = bets[k] + (bets[k]*BLACKJACK);
    cout << "Your winnings are: $" << bets[k] << endl;
}
//Dealer and Player got blackjack
else if(sum1 == 21 && sum2 == 21){
    cout << "The total value of your hand is " << sum1 << ".\n";
    cout << "You and the dealer both got Blackjack!\n"
         << "You get your money back with no damage done.\n";
}
//Dealer wins by blackjack
else if(sum2 == 21 && sum1 < sum2){
    cout << "The dealer's hand is valued " << sum2 << ".\n"
         << "The dealer got BLACKJACK\n"
         << "Sorry, you lose, better luck next time.\n";
    }
}

//Bubble sort players hands least to greatest
void bublSrt(int sum1[]){
    bool swap;
    do{
        for(int x = 0; x < SIZE - 1; x++){
            swap = false;
            if(sum1[x] > sum1[x+1]){       //Put the smallest at top of List
                sum1[x] = sum1[x]^sum1[x+1];  //In place Swap
                sum1[x+1] = sum1[x]^sum1[x+1];  //In place Swap
                sum1[x] = sum1[x]^sum1[x+1];  //In place Swap
            swap = true;
            }
        }
    }while(swap);
}

void selcSrt(int sum1[]){
    for(int i = 0; i < SIZE - 1; i++){      //Loop for each position in List
        int min=i;
        for(int j=i+1;j<SIZE;j++){  //Loop to swap with first in List
            if(sum1[min]>sum1[j]){  //Put the smallest at top of List
                min=j;
            }
        }
        int temp=sum1[i];
        sum1[i]=sum1[min];
        sum1[min]=temp;
    }
}

bool linSrch(int sum1[], int toChs){
    for(int i = 0; i < SIZE; i++){
        if(toChs == sum1[i])return true;
    }
    return false;
}

// Function to print an array
void prntArr(int sum1[]){
    for (int i = 0; i < SIZE; i++){
        cout << sum1[i] << " ";
        cout << endl;
    }
}
```