Project 2

<Casino>

CIS-5

Name: Aleksandar Videv

TABLE OF CONTENTS

**1. Summary…………………………………….………………………3**

1.1. How the Card Game Works**……………………………….…………3**

1.2. Rules of the game……………**…………………...…………………..3**

**3.The logic behind the game**……**…………………………….……...5**

**3.1. Flowchart**……**………………………………………………….……5**

3.2. Pseudo Code……**…………………………..……………………..20**

4. Program……**………………………………**……**………………..…37**

5. Proof of a working product……**……………………………….…**

**1. Summary**

Project size: about 1280+ lines

The number of variables: about 16

My casino consists of 3 games Craps, Roulette, and Random numbers. Every game has different bets you can choose from, and you will be allowed to play as long as you have the chips to play.

Also, about the flowchart and pseudo code I just did all the stuff for the craps game. From how long they are I think that should be good enough.

**1.1. How the Card Game Works**

**Craps -**In these craps, you have the main bets as an option Pass line and don't pass line. But while making bets, depending on if you get the point number, you would be allowed to make an odds bet. You are also entitled to make a come bet.

**Roulette** – you are displayed with the table and all the numbers. You can bet on

Single number bet, Column, Red/Black, and Even/Odd.

**Random numbers** – here, you can make a bet by entering ten numbers between 10000 and 0.

**1.2 Rules of the game**

**Crabs**

**Pass Line Bet**

The most popular craps bet, the Pass Line, has a house edge of just 1.41% and is the most straightforward wager to make. A pass line bet means players wager on the shooter landing a 7 or 11 before a 2, 3, or 12 on the come-out roll. If the dice land on 7 or 11, the pass bet wins; if it lands on 2, 3, or 12, it loses. If it rolls on any other number, the pass bet stays active on the craps table until the shooter lands the point number again, or a 7.

**Don't Pass Line Bet**

This is an inverted bet that can be played at most times during a game. It is a reversal of the Pass Line bet. Don't Pass Lines will see players win on a 2 or 3 roll, while a 7 or 11 will lose. A throw of 12 means the bet is a push, and any other numbers must not reappear before a 7 for the player to win. While the lower house edge on this bet may be appealing, it's an unpopular bet at the craps table because the player is essentially betting on other gamblers to lose.

**Come Bet**

Players can only make this bet after the point on the Pass Line has been determined, but the rules are precisely the same as those for a Pass Line bet. After the point has been set, if a player makes a come bet, it means they're betting on the shooter landing a 7 or 11 on the dice before they come to the point.

**Odds Bets**

Players can only 'lay the odds' after a pass or bet. On this wager, players wager the shooter will land a seven before a specific point number is thrown. As this bet is statistically fair, these wagers have no house edge. Little wonder then that these are popular bets with craps players.

**Roulette**

**Single Number Bets (also known as Straight-up Bets or Classic Bets)**:

Players may wager on any individual number by placing chips on top of that number. If the number is hit on the next spin, this wager pays 35-1.

**Column Bets:**

This bet covers one of the three columns of 12 numbers found on the roulette wheel. If any number in that column wins, the bet pays 2-1.

**Red/Black Bets:**

These two bets cover all numbers with pockets of the appropriate color on the roulette wheel. If any number of that color wins, the bet pays even money.

**Odd/Even Bets**:

These two bets cover all odd numbers or even numbers, respectively. If any number of that type wins, this bet will pay even money.

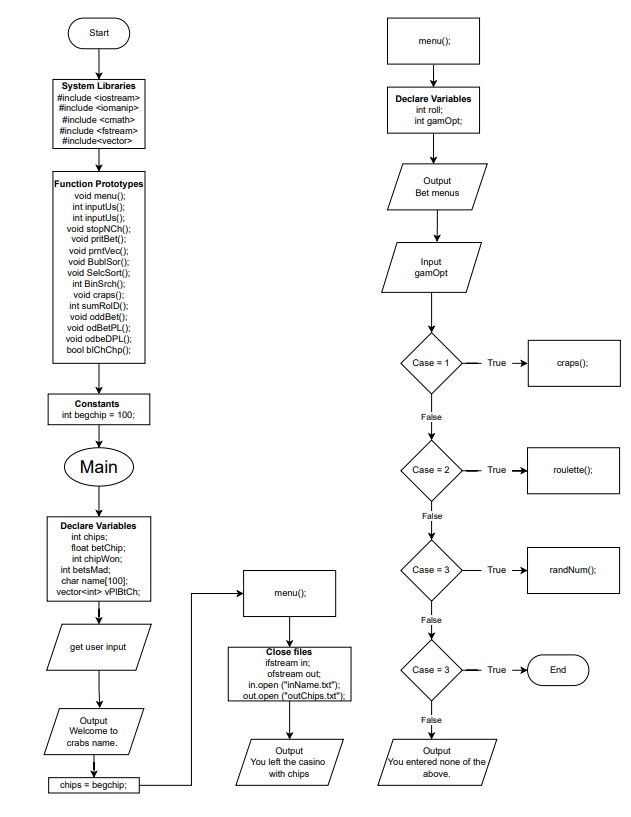
**Random numbers**

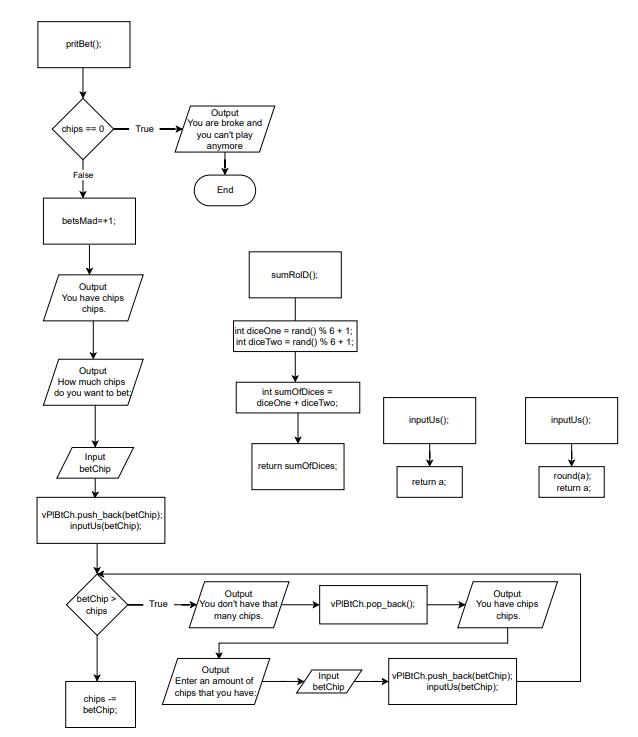
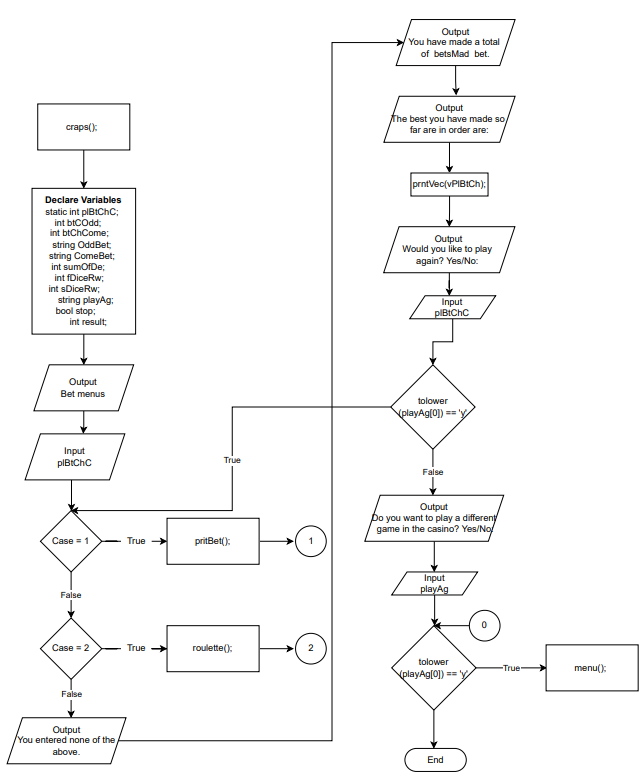
You enter ten numbers between 10000 and 0, and if they match the same random numbers, you win your bet with an x100 return.

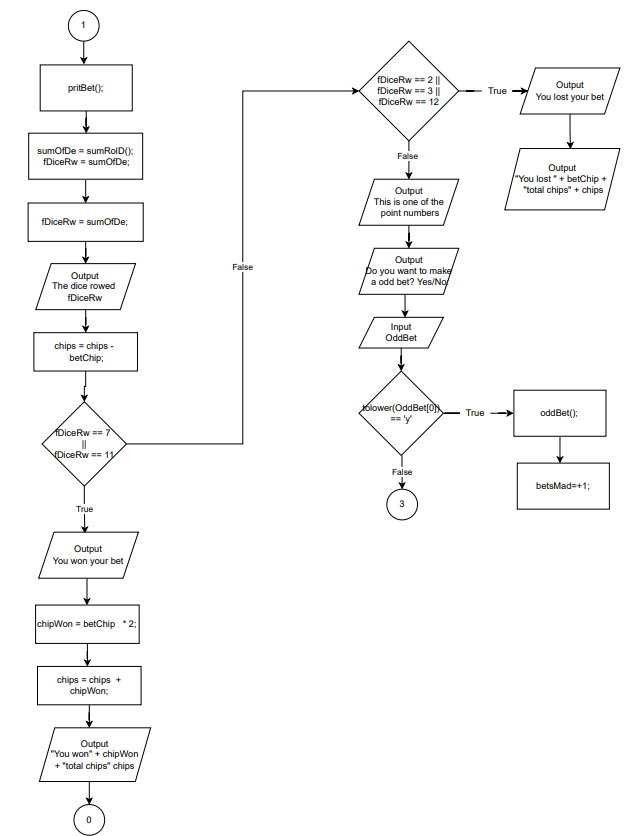
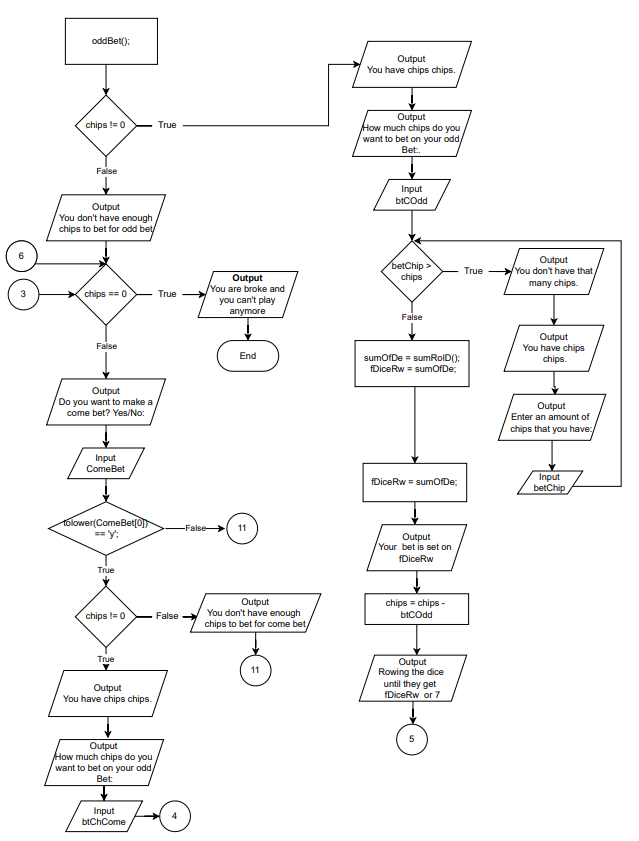
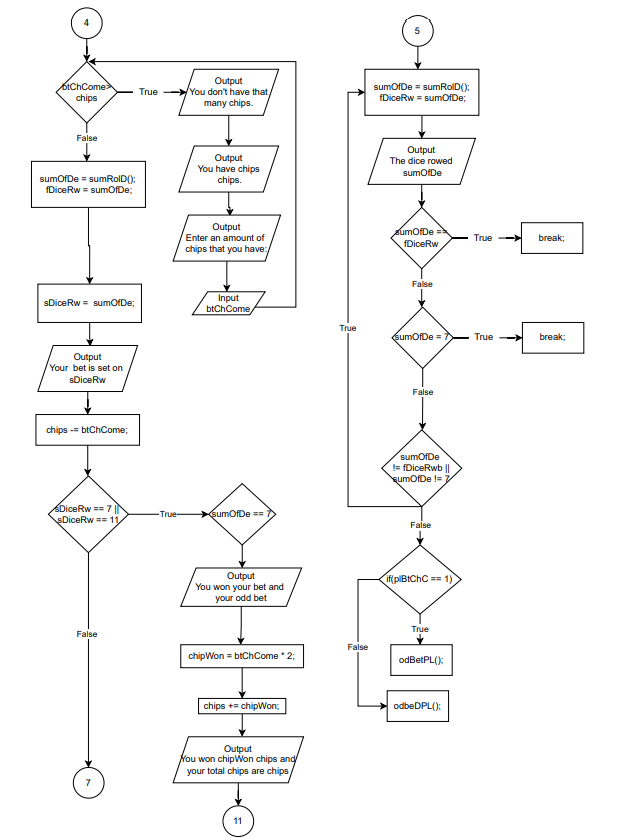
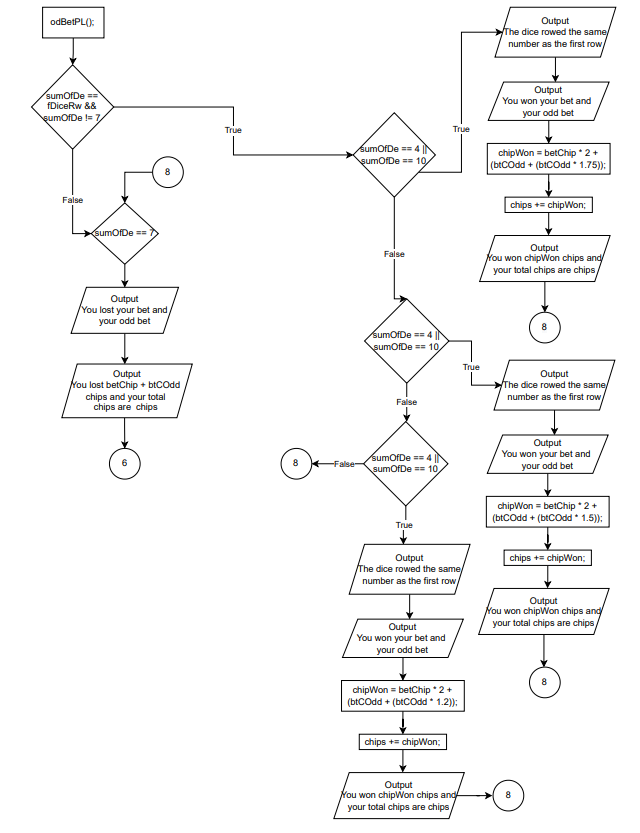
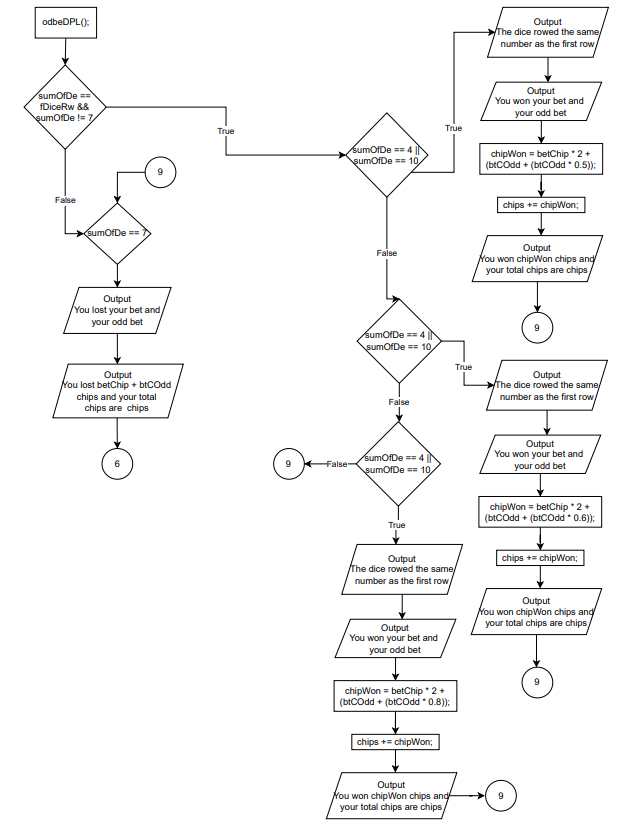
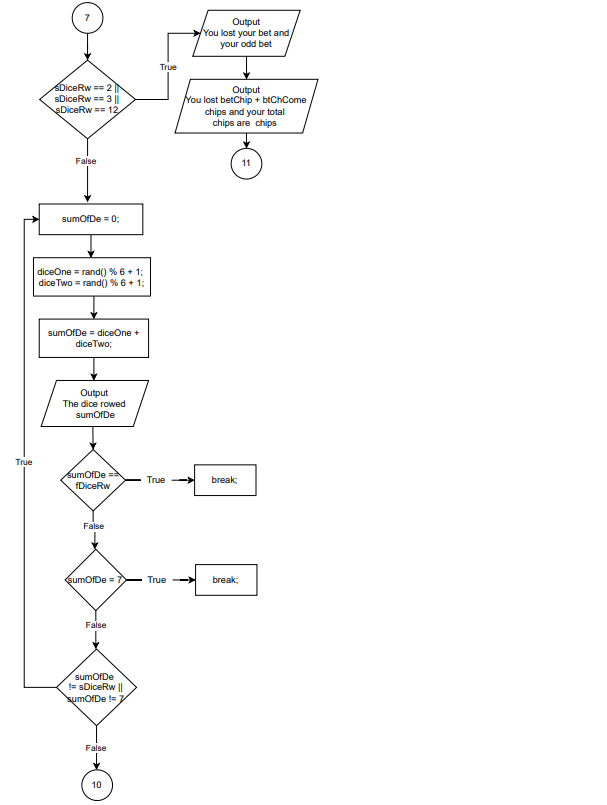
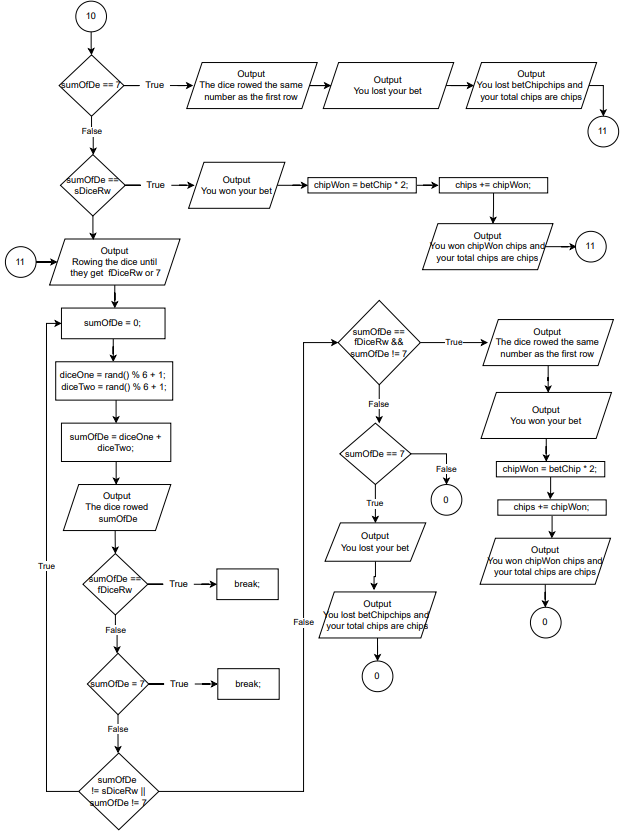
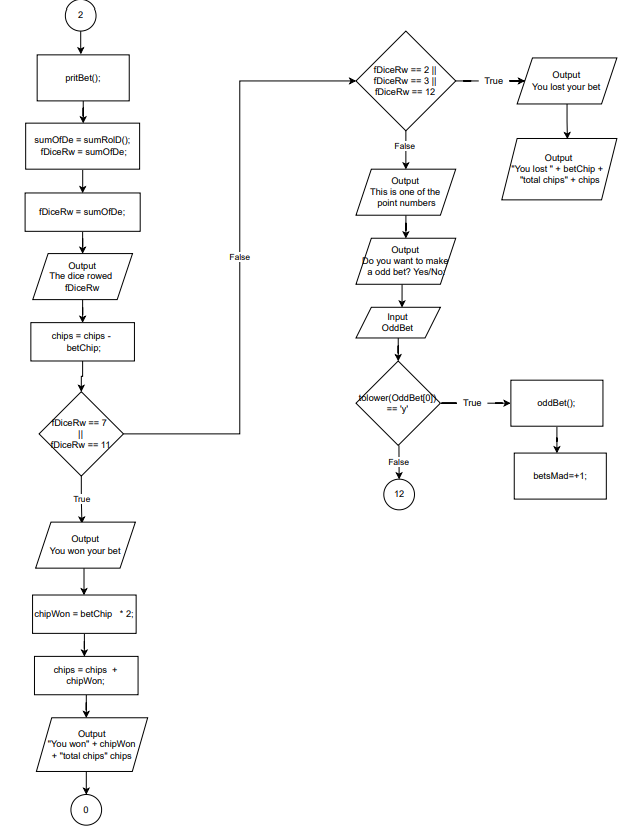
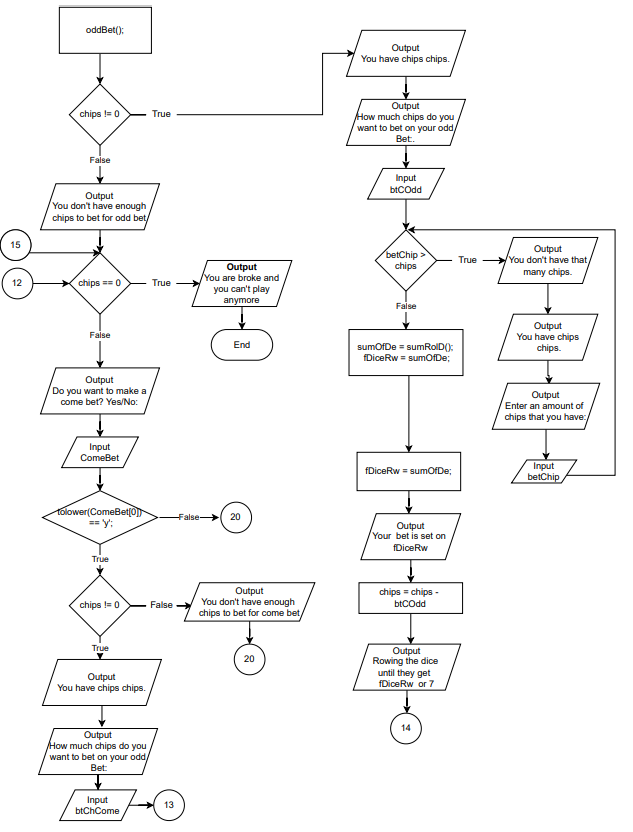
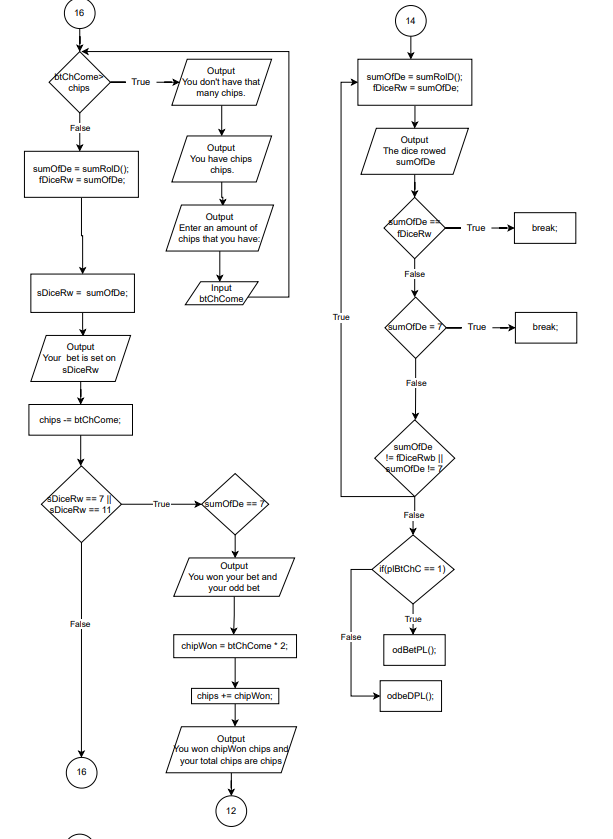
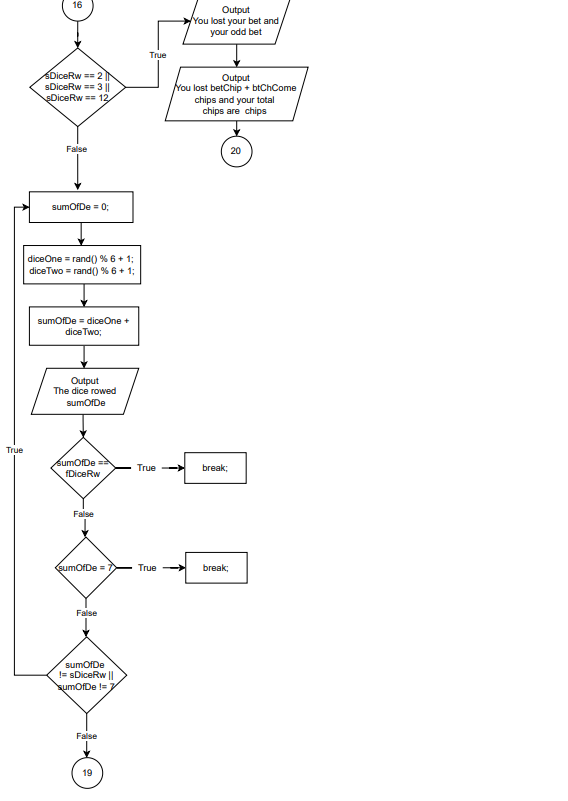
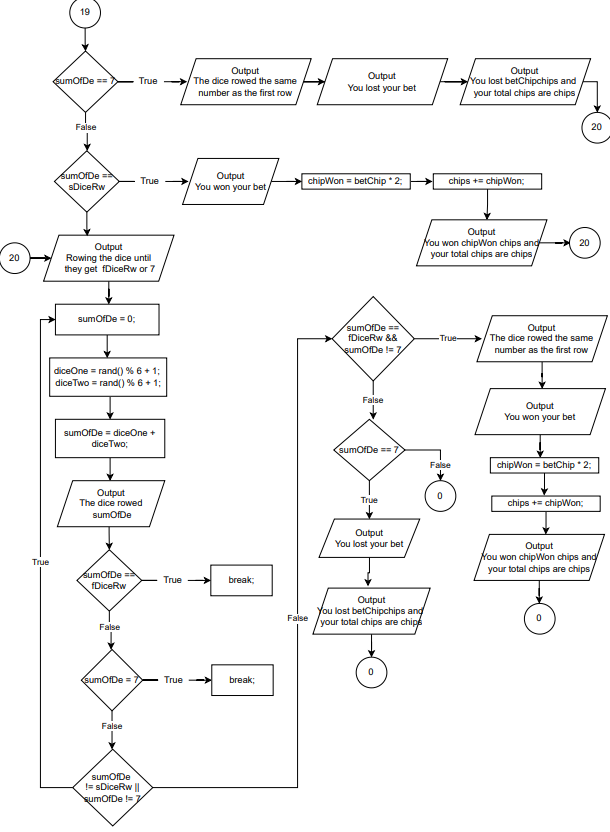
Source: <https://www.onlinegambling.com/casino/craps/bets/>

<https://www.rouletteonline.net/roulette-bets/>

**3.The logic behind the game**

**3.1. Flowchart** 

**3.2. Pseudo Code**

Create Libraries

add Function Prototypes

add const int begchip = 100;

main

Declare Variables

int chips;

float betChip = 0;

int chipWon = 0;

int betsMad = 0;

char name[100];

vector<int> vPlBtCh;

open files

get the input from the files

print the file input in the program

chips = begchip;

call funtion menu(chimenups, betChip, chipWon, betsMad, vPlBtCh);

out << "You left the casino with " + chips + "chips";

///Exit the code

in.close();

out.close();

function menu

Declare Variables

int roll;

int gamOpt;

print bets

cin >> gamOpt;

choose from

1

call funtion craps(chips, betChip, chipWon, betsMad, vPlBtCh);

choose from

2

call funtion roulette( chips, betChip, chipWon, betsMad, vPlBtCh);

choose from

3

call funtion randNum(chips, betChip, chipWon, betsMad, vPlBtCh);

choose from

4

break

function pritBet(int &chips, float &betChip, int &betsMad, vector<int> &vPlBtCh)

if (chips == 0)

print You are broke and you can't play anymore

end program

betsMad=betsMad+1;

print How much chips do you want to bet

input betChip

vPlBtCh.push\_back(betChip);

inputUs(betChip);

while(betChip > chips)

vPlBtCh.pop\_back();

print You don't have that many chips.

print How much chips do you want to bet

input betChip

vPlBtCh.push\_back(betChip);

inputUs(betChip);

chips =chips - betChip;

function sumRolD

int diceOne = rand() % 6 + 1; // get a rand num between 1 and 6

int diceTwo = rand() % 6 + 1; // get a rand num between 1 and 6

int sumOfDices = diceOne + diceTwo;

return sumOfDices;

function craps

print bets

choose from

1

if (chips == 0)

pritBet(chips, betChip, betsMad, vPlBtCh);

firstDiceRow = sumRolD();

print the dice rowed firstDiceRow

chips = chips - betChip;

if(firstdice = 7 or firstdice == 11)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon

elseif(firstdice = 2 or firstdice = 3 or firstdice = 12)

print You lost your bet

else

// point numbers , odd bets, come bets

print This is one of the point numbers

print Do you want to make a odd bet? Yes/No:

input doOddBet

if(doOddBet = 'Yes' or 'y')

if(chips is different then 0)

print How much chips do you want to bet on your odd Bet:

input betchisonOdd

while(betChisOnOdd > chips)

print You don't have that many chips.

print How much chips do you want to bet

input betchisonOdd

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

firstDiceRow = sumOfDice;

print Your bet is set on firstDiceRow

print Rowing the dice until they get firstDiceRow or 7

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to firstDiceRow or sumOfDice is not equal to 7)

if(sumOfDice = firstDiceRow and sumOfDice is not equal to 7)

if(sumOfDice = 4 or sumOfDice = 10)

print The dice rowed the same number as the first row

print You won your bet and your odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 1.75));

chips = chips + chipWon;

else if(sumOfDice = 5 or sumOfDice = 9)

print The dice rowed the same number as the first row

print You won your bet and your odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 1.5));

chips = chips + chipWon;

else

print The dice rowed the same number as the first row

print You won your bet and your odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 1.2));

chips = chips + chipWon;

if(sumOfDice = 7)

print You lost your bet and your odd bet

else

print You don't have enough chips to bet for odd bet

if(chips = 0)

print You are broke and you can't play anymore

print "Do you want to make a come bet? Yes/No:

input doComeBet

for(ComeBet; doComeBet = 'Yes' or 'y')

if(chips is different then 0)

print How much chips do you want to bet on your odd Bet:

input betchisonOdd

while(betChisOnOdd > chips)

print You don't have that many chips.

print How much chips do you want to bet

input betchisonOdd

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

secondDiceRow = sumOfDice;

print Your bet is set on secondDiceRow

chips = chips - betchisonOdd;

if (secondDiceRow = 7 or secondDiceRow = 11)

print You won your bet

chipWon = betchisonOdd \* 2;

chips = chips + chipWon;

else if(secondDiceRow = 2 or secondDiceRow = 3 or secondDiceRow = 12)

print You lost your bet

else

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to secondDiceRow or sumOfDice is not equal to 7)

if(sumOfDe = 7)

print The dice rowed the same number as the first row

print You lost your bet

if(sumOfDe = secondDiceRow)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon;

else

print You don't have enough chips to bet for odd bet

stop program

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to firstDiceRow or sumOfDice is not equal to 7)

if(sumOfDe = fDiceRw and sumOfDe is not equal to 7)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon;

if(sumOfDe = 7)

print You lost your bet

stop the loop

2

pritBet(chips, betChip, betsMad, vPlBtCh);

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo

firstDiceRow = sumOfDice;

print the dice rowed firstDiceRow

chips = chips - betChip;

if(firstdice = 2 or firstdice == 3)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon

elseif(firstdice = 7 or firstdice = 1 )

print You lost your bet

else if (firstdice = 12)

print It is a tie

chipWon = betChip;

chips = chips + chipWon

else

// point numbers , odd bets, come bets

print This is one of the point numbers

print Do you want to make a odd bet? Yes/No:

input doOddBet

if(doOddBet = 'Yes' or 'y')

if(chips is different then 0)

print How much chips do you want to bet on your odd Bet:

input betchisonOdd

while(betChisOnOdd > chips)

print You don't have that many chips.

print How much chips do you want to bet

input betchisonOdd

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

firstDiceRow = sumOfDice;

print Your bet is set on firstDiceRow

print Rowing the dice until they get firstDiceRow or 7

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to firstDiceRow or sumOfDice is not equal to 7)

if(sumOfDice = firstDiceRow and sumOfDice is not equal to 7)

if(sumOfDice = 4 or sumOfDice = 10)

print The dice rowed the same number as the first row

print You lost your , but you won the odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 0.6));

chips = chips + chipWon;

else if(sumOfDice = 5 or sumOfDice = 9)

print The dice rowed the same number as the first row

print You lost your , but you won the odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 0.8));

chips = chips + chipWon;

else

print The dice rowed the same number as the first row

print You lost your , but you won the odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 0.5));

chips = chips + chipWon;

if(sumOfDice = 7)

print You won your bet and your odd bet

chipWon = betChip \* 2 + (betchisonOdd + (betchisonOdd \* 1.75));

chips = chips + chipWon;

else

print You don't have enough chips to bet for odd bet

if(chips = 0)

print You are broke and you can't play anymore

print "Do you want to make a come bet? Yes/No:

input doComeBet

for(ComeBet; doComeBet = 'Yes' or 'y')

if(chips is different then 0)

print How much chips do you want to bet on your odd Bet:

input betchisonOdd

while(betChisOnOdd > chips)

print You don't have that many chips.

print How much chips do you want to bet

input betchisonOdd

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

secondDiceRow = sumOfDice;

print Your bet is set on secondDiceRow

chips = chips - betchisonOdd;

if (secondDiceRow = 7 or secondDiceRow = 11)

print You won your bet

chipWon = betchisonOdd \* 2;

chips = chips + chipWon;

else if(secondDiceRow = 2 or secondDiceRow = 3 or secondDiceRow = 12)

print You lost your bet

else

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to secondDiceRow or sumOfDice is not equal to 7)

if(sumOfDe = 7)

print The dice rowed the same number as the first row

print You lost your bet

if(sumOfDe = secondDiceRow)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon;

else

print You don't have enough chips to bet for odd bet

stop program

do

sumOfDice = 0;

row dice one = Get a rand num between 1 and 6 for dice

row dice two = Get a rand num between 1 and 6 for dice

sumOfDice = diceOne + diceTwo;

print The dice rowed sumOfDice

while(sumOfDice is not equal to firstDiceRow or sumOfDice is not equal to 7)

if(sumOfDe = fDiceRw and sumOfDe is not equal to 7)

print You won your bet

chipWon = betChip \* 2;

chips = chips + chipWon;

if(sumOfDe = 7)

print You lost your bet

stop the loop

**4. Program**

/\*

\* File: main.cpp

\* Author: Aleksandar Videv

\* Created: 5/09/2022

\* Purpose: Casino

\*

\*/

//System Libraries

#include <iostream>

#include <iomanip>

#include <cmath>

#include <fstream>

#include<vector>

using namespace std;

//Function Prototypes

void menu(int, float, int, int, vector<int> &); //Print Menu

int inputUs(int); //Returns a

int inputUs(float); //Overloading

void stopNCh(int); //Stops the program

void pritBet(int &, float &, int &, vector<int> &); //Check if the user bet is possible

void prntVec(vector<int> &vPlBtCh); //Print vector

void BublSor(int [], int); //Bubble Sort

void SelcSort(int [], int); //Selection Sort

int BinSrch(int [], int, int); //Binary Search

//Craps game

void craps(int, float, int, int &, vector<int> &); //Craps

int sumRolD(); // Sum of dice

void oddBet(int, int, int, int, int, float, int); //Odd bet

void odBetPL(int , int , int , int , float , int ); //Odd bet inside the pass line

void odbeDPL(int , int , int , int , float , int ); //Odd bet inside the Don't pass line

bool blChChp(int); //Bool check chips

//Roulette game

void roulette(int, float, int, int, vector<int> &); //Roulette

void spnRot(int, int [], int [], int [], int [],int [],int [],int [], int [], string []); // Spin the Roulette

void prntRot(); // Print the Roulette

// Roulette array bets

//Random numbers game

void randNum(int, float, int, int , vector<int> &vPlBtCh); //Random numbers

//Global Constants, no Global Variables are allowed

const int begchip = 100; // Begging Chips

//Execution Begins Here!

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

{

srand(static\_cast<unsigned int>(time(0))); // Allows us to get a random number by creating time

//Declare Variables

int chips; //Every player starts with 100 chips

float betChip = 0; //How many of the chips he want to bet

int chipWon = 0; //How many chis did the player win

int betsMad = 0; //How many bets has the player made

char name[100]; //Takes a name the size of 100 letters

vector<int> vPlBtCh; //Vector that stores the player bets

ifstream in;

ofstream out;

in.open ("inName.txt");

out.open ("outChips.txt");

//Initialize or input i.e. set variable values

//Display the outputs

in >> name;

cout << "Welcome the casino " << name << "." << endl;

chips = begchip;

menu(chips, betChip, chipWon, betsMad, vPlBtCh);

out << "You left the casino with " << chips << "chips";

//Exit the code

in.close();

out.close();

return 0;

}

//Print Menu

void menu(int chips, float betChip, int chipWon, int betsMad, vector<int> &vPlBtCh)//Print Menu

{

int roll; // Stores the roll of the roulette

int gamOpt; // Player choice for game

cout << "1. Craps" << endl;

cout << "2. Roulette" << endl;

cout << "3. Random numbers" << endl;

cout << "3. EXIT" << endl;

cout << "Which game do you want to play:" << endl;

cin >> gamOpt;

switch(gamOpt)

{

case 1:

cout << " " << endl;

craps(chips, betChip, chipWon, betsMad, vPlBtCh);

break;

case 2:

cout << " " << endl;

roulette( chips, betChip, chipWon, betsMad, vPlBtCh);

break;

case 3:

cout << " " << endl;

randNum(chips, betChip, chipWon, betsMad, vPlBtCh);

break;

case 4:

break;

break;

default:

cout << "You entered none of the above." << endl;

}

}

//Check if you have more then begging chips

bool blChChp (int x = 0)

{

return (x > 100);

}

//If you have played more then 10 games the casino gives 1 chip for free

//Returns a if it is an int

int inputUs(int a)

{

return a;

}

//Overloading and return a rounded up a if input is with decimals

int inputUs(float a)

{

round(a);

return a;

}

//Stops the program if you don't have any more chips

void stopNCh(int ch)

{

if(ch <= 0)

{

cout << "You are out of chips and you can't bet anymore";

exit(0);

}

}

//Print vector

void prntVec(vector<int> &vPlBtCh)

{

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

{

cout << vPlBtCh[i] << " ";

}

}

//Check if the user bet is possible

void pritBet(int &chips, float &betChip, int &betsMad, vector<int> &vPlBtCh)

{

if(chips == 0) // Checks for the user has more then 0 chis to play

{

cout << " " << endl;

cout << "You are broke and you can't play anymore" << endl;

exit(0);

}

betsMad=+1;

cout << " " << endl;

cout << "You have " << chips << " chips." << endl;

cout << "How much chips do you want to bet:" << endl;

cin >> betChip;

vPlBtCh.push\_back(betChip);

inputUs(betChip);

while(betChip >= chips) // Doesn't let the user to input more chips than he has

{

vPlBtCh.pop\_back();

cout << " " << endl;

cout << "You don't have that many chips." << endl;

cout << "You have " << chips << " chips." << endl;

cout << "Enter an amount of chips that you have:" << endl;

cin >> betChip;

vPlBtCh.push\_back(betChip);

inputUs(betChip);

}

chips -= betChip;

}

//Bubble sort

void BublSor(int randNum[], int size)

{

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

{

for (int j = 0; i < (size - j - 1); j++)

{

if(randNum[j]> randNum[j+1])

{

int temp = randNum[j] ;

randNum[j] = randNum[j + 1 ] ;

randNum[j + 1] = temp ;

}

}

}

cout << "Here are all the random numbers sorted with Bubble Sort:" << endl;

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

{

cout<<randNum[i]<<" ";

}

cout << "" << endl;

}

//Selection Sort

void SelcSort(int randNum[], int size)

{

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

{

int min = i;

for (int j = i + 1; j < size; j++)

{

if (randNum[j] < randNum[min])

{

min = j;

}

}

if (min != i)

{

int temp = randNum[min];

randNum[min] = randNum[i];

randNum[i] = temp;

}

}

cout << "Here are all the random numbers sorted with Selection Sort:" << endl;

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

{

cout<<randNum[i]<<" ";

}

cout << "" << endl;

}

//Binary Search

int BinSrch(int randNum[], int size, int userVal)

{

int low = 0;

int high = size - 1;

int mid;

while (low <= high)

{

mid = (low + high) / 2;

if(userVal == randNum[mid])

{

return mid;

}

else if (userVal > randNum[mid])

{

low = mid + 1;

}

else

{

high = mid - 1;

}

}

return -1;

}

//Print craps

void craps(int chips, float betChip, int chipWon, int &betsMad, vector<int> &vPlBtCh)

{

static int plBtChC; // What bet does the player want to make for craps

int btCOdd; // Bet chips on odd

int btChCome; // Bet chips on come

string OddBet; // If he want to make an odd bet

string ComeBet; // If he want to make an come bet

int sumOfDe; // The sum of both Dice

int fDiceRw; // The first sum the the rowed dice

int sDiceRw; // The second sum the the rowed dice

string playAg; // Ask if the player want to play again

bool stop; // Stops the program

int result;

// Loop until the user doesn't want to play

do

{

cout << "You can bet on:" << endl;

cout << "1. Pass line" << endl;

cout << "2. Don’t pass line" << endl;

cout << "What bet do you want to make of the following:" << endl;

cin >> plBtChC;

switch(plBtChC)

{

// Pass Line

case 1:

pritBet(chips, betChip, betsMad, vPlBtCh);

sumOfDe = sumRolD();

fDiceRw = sumOfDe;

cout << " " << endl;

cout << "The dice rowed " << fDiceRw << endl;

// Checking if the rolled dice are equal to 7, 11

if (fDiceRw == 7 || fDiceRw == 11)

{

cout << " " << endl;

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to 2, 3, 12

else if (fDiceRw == 2 || fDiceRw == 3 || fDiceRw == 12)

{

cout << " " << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

// Go to point numbers and ask him if he want to make an odd or come bet

else

{

cout << "This is one of the point numbers" << endl;

cout << " " << endl;

cout << "Do you want to make a odd bet? Yes/No:" << endl;

cin >> OddBet;

if(tolower(OddBet[0]) == 'y') // If the user input starts with Y or y it would go inside

{

odBetPL(btCOdd, sumOfDe, fDiceRw, chips, betChip, chipWon);

betsMad=+1;

}

// Checks for the user has more then 0 chis to play

if(chips == 0)

{

cout << " " << endl;

cout << "You are broke and you can't play anymore" << endl;

exit(0);

}

cout << "Do you want to make a come bet? Yes/No:" << endl;

cin >> ComeBet;

for(ComeBet; tolower(ComeBet[0]) == 'y';) // If the user input starts with Y or y it would go inside

{

betsMad=+1;

// Checks for the user has more then 0 chis to play

if(chips != 0)

{

cout << "You have " << chips << " chips." << endl;

cout << "How much chips do you want to bet on your come bet:" << endl;

cin >> btChCome;

// Doesn't let the user to input more chips than he has

while(btChCome > chips)

{

cout << "You don't have that many chips." << endl;

cout << "You have " << chips << " chips." << endl;

cout << "Enter an amount of chips that you have:" << endl;

cin >> btChCome;

}

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

sDiceRw = sumOfDe;

cout << "Your come bet is set on " << sDiceRw << endl;

chips -= btChCome;

// Checking if the rolled dice are equal to 7 or 11

if (sDiceRw == 7 || sDiceRw == 11)

{

cout << "You won your bet" << endl;

chipWon = btChCome \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to 2, 3, 12

else if (sDiceRw == 2 || sDiceRw == 3 || sDiceRw == 12)

{

cout << "You lost your bet" << endl;

cout << "You lost " << btChCome << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to every other number

else

{

do

{

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

cout << "The dice rowed " << sumOfDe << endl;

if(sumOfDe == fDiceRw) // Stop the loop when dice get the same number as the first one

{

break;

}

if(sumOfDe == 7) // Stop the loop when dice get 7

{

break;

}

// Loop until the dice rows 7 or the first number

}while(sumOfDe != sDiceRw || sumOfDe != 7);

// Checks if the first row is equal to 7

if(sumOfDe == 7)

{

cout << "The dice rowed the same number as the first row" << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

// Checks if the first row is second row is equal to the first one

if(sumOfDe == sDiceRw)

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

}

}

else

{

cout << "You don't have enough chips to bet for come bet" << endl;

}

break;

}

cout << " " << endl;

cout << "Rowing the dice until they get " << fDiceRw << " or 7" << endl;

do

{

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

cout << "The dice rowed " << sumOfDe << endl;

// Stops the loop when second row is equal to the first one

if(sumOfDe == fDiceRw)

{

break;

}

// Stop the loop first row is equal to 7

if(sumOfDe == 7)

{

break;

}

// Loop until the dice rows 7 or the first row

}while(sumOfDe != fDiceRw || sumOfDe != 7);

// // Checks if the first row is second row is equal to the first one

if(sumOfDe == fDiceRw && sumOfDe != 7)

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checks if the first row is equal to 7

if(sumOfDe == 7)

{

cout << " " << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

}

break;

// Don’t pass bets

case 2:

pritBet(chips, betChip, betsMad, vPlBtCh);

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

fDiceRw = sumOfDe;

cout << " " << endl;

cout << "The dice rowed " << fDiceRw << endl;

// Checking if the rolled dice are equal to 2, 3

if (fDiceRw == 2 || fDiceRw == 3)

{

cout << " " << endl;

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to 7, 11

else if (fDiceRw == 7 || fDiceRw == 11)

{

cout << " " << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to 12

else if (fDiceRw == 12)

{

cout << " " << endl;

cout << "It is a tie. You keep your bet." << endl;

chipWon = betChip;

chips += chipWon;

cout << "You have " << chips << " chips." << endl;

}

// Go to point numbers and ask him if he want to make an odd or come bet

else

{

cout << "This is one of the point numbers" << endl;

cout << " " << endl;

cout << "Do you want to make a odd bet? Yes/No:" << endl;

cin >> OddBet;

if(tolower(OddBet[0]) == 'y') //// If the user input starts with Y or y it would go inside

{

// Checks for the user has more then 0 chis to play

odBetPL(btCOdd, sumOfDe, fDiceRw, chips, betChip, chipWon);

betsMad=+1;

}

// Checks if the user has more chips then 0 after the odd bet. If yes he can continue playing.

if(chips == 0)

{

stop = false;

if(stop == false)

{

exit(0);

}

cout << " " << endl;

cout << "You are broke and you can't play anymore" << endl;

break;

}

cout << "Do you want to make a come bet? Yes/No:" << endl;

cin >> ComeBet;

for(ComeBet; tolower(ComeBet[0]) == 'y';) // If the user input starts with Y or y it would go inside

{

betsMad=+1;

if(chips != 0)

{

cout << "You have " << chips << " chips." << endl;

cout << "How much chips do you want to bet on your come bet:" << endl;

cin >> btChCome;

// Will lope if the user input more chips then he has.

while(btChCome > chips)

{

cout << "You don't have that many chips." << endl;

cout << "You have " << chips << " chips." << endl;

cout << "Enter an amount of chips that you have:" << endl;

cin >> btChCome;

}

if(chips == 0)

{

for (int i = 0; i < 1; i++) {

cout << " " << endl;

cout << "You are broke and you can't play anymore" << endl;

break;

}

}

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

sDiceRw = sumOfDe;

cout << "Your come bet is set on " << sDiceRw << endl;

chips -= btChCome;

if (sDiceRw == 7 || sDiceRw == 11)

{

cout << "You won your bet" << endl;

chipWon = btChCome \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

chips > begchip ? "Wow, you have more chips than you had in the beginning. Good job.\n" : "" ;

}

// Checking if the rolled dice are equal to 2, 3, 12

else if (sDiceRw == 2 || sDiceRw == 3 || sDiceRw == 12)

{

cout << "You lost your bet" << endl;

cout << "You lost " << btChCome << " chips and your total chips are " << chips << endl;

}

else

{

do

{

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

cout << "The dice rowed " << sumOfDe << endl;

// Stop the loop when dice get the same number as the first one

if(sumOfDe == sDiceRw)

{

break;

}

// Stop the loop when dice get 7

if(sumOfDe == 7)

{

break;

}

// Loop until the dice rows 7 or the first number

}while(sumOfDe != sDiceRw || sumOfDe != 7);

// Checks if the number that stop the top loop 7

if(sumOfDe == 7)

{

cout << "The dice rowed the same number as the first row" << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

// Checks if the number that stop the top loop is same number as the first one

if(sumOfDe == sDiceRw)

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

}

}

else

{

cout << "You don't have enough chips to bet for come bet" << endl;

}

break;

}

result = pow(betChip, chipWon);

cout << " " << endl;

cout << "Rowing the dice until they get " << fDiceRw << " or 7" << endl;

do

{

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

cout << "The dice rowed " << sumOfDe << endl;

// Stop the loop when dice get the same number as the first one

if(sumOfDe == fDiceRw)

{

break;

}

// Stop the loop when dice get 7

if(sumOfDe == 7)

{

break;

}

// Loop until the dice rows 7 or the first number

}while(sumOfDe != fDiceRw || sumOfDe != 7);

// Checks if the number that stop the top loop 7

if(sumOfDe == 7)

{

cout << " " << endl;

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checks if the number that stop the top loop is same number as the first one

if(sumOfDe == fDiceRw && sumOfDe != 7)

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

if(blChChp (chips) == '1')

{

cout <<"Good job you have more chis then you started with" << endl;

}

}

stopNCh(chips);

break;

default:

cout << "You entered none of the above." << endl;

}

cout << "You have made a total of " << betsMad << " bet." << endl;

cout << "The best you have made so far are in order are:" << endl;

prntVec(vPlBtCh);

cout << " " << endl;

cout << "Would you like to play again? Yes/No: " << endl;

cin >> playAg;

cout << " " << endl;

// Checks if the first letter of the input of the user is Y/y. if yes do again.

}while(tolower(playAg[0]) == 'y');

cout << "Do you want to play a different game in the casino? Yes/No:" << endl;

cin >> playAg;

if(tolower(playAg[0]) == 'y')

{

menu(chips, betChip, chipWon, betsMad, vPlBtCh);

}

}

// Roll the two dice and gets the sum of both of them

int sumRolD()

{

int diceOne = rand() % 6 + 1; // get a rand num between 1 and 6

int diceTwo = rand() % 6 + 1; // get a rand num between 1 and 6

int sumOfDices = diceOne + diceTwo;

return sumOfDices;

}

//Creates an odd bet

void oddBet(int plBtChC, int btCOdd, int sumOfDe, int fDiceRw, int chips, float betChip, int chipWon)

{

if(chips != 0) // Checks for the user has more then 0 chis to play

{

cout << "You have " << chips << " chips." << endl;

cout << "How much chips do you want to bet on your odd Bet:" << endl;

cin >> btCOdd;

cout << " " << endl;

while(btCOdd > chips) // Doesn't let the user to input more chips than he has

{

cout << "You don't have that many chips." << endl;

cout << "You have " << chips << " chips." << endl;

cout << "Enter an amount of chips that you have:" << endl;

cin >> btCOdd;

cout << " " << endl;

}

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

fDiceRw = sumOfDe;

cout << "Your bet is set on " << fDiceRw << endl;

chips -= btCOdd;

cout << " " << endl;

cout << "Rowing the dice until they get " << fDiceRw << " or 7" << endl;

// Loop until the dice rows 7 or the first number

do

{

sumOfDe = sumRolD(); // Roll the two dice and gets the sum of both of them

cout << "The dice rowed " << sumOfDe << endl;

if(sumOfDe == fDiceRw) // Stop the loop when dice get the same number as the first one

{

break;

}

if(sumOfDe == 7) // Stop the loop when dice get 7

{

break;

}

}while(sumOfDe != fDiceRw || sumOfDe != 7);

// Odd bet on the pass line

if(plBtChC == 1)

{

odBetPL(btCOdd, sumOfDe, fDiceRw, chips, betChip, chipWon);

}

else

{

odbeDPL(btCOdd, sumOfDe, fDiceRw, chips, betChip, chipWon);

}

}

else

{

cout << "You don't have enough chips to bet for odd bet" << endl;

}

}

//Creates an odd bet inside the pass line

void odBetPL(int btCOdd, int sumOfDe, int fDiceRw, int chips, float betChip, int chipWon)

{

if(sumOfDe == fDiceRw && sumOfDe != 7)

{

// Checking if the rolled dice are equal to 4 or 10

if(sumOfDe == 4 || sumOfDe == 10)

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You won your bet and your odd bet" << endl;

chipWon = betChip \* 2 + (btCOdd + (btCOdd \* 1.75));

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to 5 or 9

else if(sumOfDe == 5 || sumOfDe == 9)

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You won your bet and your odd bet" << endl;

chipWon = betChip \* 2 + (btCOdd + (btCOdd \* 1.5));

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

// Checking if the rolled dice are equal to every other number

else

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You won your bet and your odd bet" << endl;

chipWon = betChip \* 2 + (btCOdd + (btCOdd \* 1.2));

chips += chipWon;

cout << "You won " << fixed << setprecision(1) << showpoint << chipWon << " chips and your total chips are " << chips << endl;

}

}

if(sumOfDe == 7) // Checking if the rolled dice are equal to 7

{

cout << " " << endl;

cout << "You lost your bet and your odd bet" << endl;

cout << "You lost " << betChip + btCOdd << " chips and your total chips are " << chips << endl;

}

}

//Creates an odd bet inside the Don't pass line

void odbeDPL(int btCOdd, int sumOfDe, int fDiceRw, int chips, float betChip, int chipWon)

{

if(sumOfDe == fDiceRw && sumOfDe != 7)

{

if(sumOfDe == 4 || sumOfDe == 10)

{

cout << " " << endl;

cout << "You lost your , but you won the odd bet" << endl;

chipWon = btCOdd + (btCOdd \* 0.5);

chips += chipWon;

cout << "You lost " << betChip << "from your bet and won" + btCOdd << " chips from your odd bet. Your total chips are " << chips << endl;

}

else if(sumOfDe == 5 || sumOfDe == 9)

{

cout << " " << endl;

cout << "You lost your , but you won the odd bet" << endl;

chipWon = btCOdd + (btCOdd \* 0.6);

chips += chipWon;

cout << "You lost " << betChip << "from your bet and won" + btCOdd << " chips from your odd bet. Your total chips are " << chips << endl;

}

else

{

cout << " " << endl;

cout << "You lost your , but you won the odd bet" << endl;

chipWon = btCOdd + ((int)btCOdd \* 0.8);

chips += chipWon;

cout << "You lost " << betChip << "from your bet and won" + btCOdd << " chips from your odd bet. Your total chips are " << chips << endl;

}

}

// Checks if the number that stop the top loop 7

if(sumOfDe == 7)

{

cout << " " << endl;

cout << "The dice rowed the same number as the first row" << endl;

cout << "You won your bet and your odd bet" << endl;

chipWon = betChip \* 2 + (btCOdd + (btCOdd \* 1.75));

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

}

//Roulette

void roulette(int chips, float betChip, int chipWon, int betsMad, vector<int> &vPlBtCh)

{

int black[] = {2,4,6,8,10,11,13,15,17,20,22,24,26,28,29,31,33,35};

int red[] = {1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36 };

int column1[] = {1,4,7,10,13,16,19,22,25,28,31,34};

int column2[] = {2,5,8,11,14,17,20,23,26,29,32,35};

int column3[] = {3,6,9,12,15,18,21,24,27,30,33,36};

int even[] = {0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,37}; //20

int odd[] = {1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35}; //18

int green[] = {0, 37};

string color[] = {" Black ", " Red ", " Green "};

int roll; // Random roll

int plBtCh; //What bet does the player want to make

string playAg; //plAY AGAIN

int secChB; //Second choice in the bet

cout << "Starting Roulette" << endl;

cout << " " << endl;

do

{

prntRot();

cout << " " << endl;

cout << "1. Single number bet" << endl;

cout << "2. Column" << endl;

cout << "3. Red/Black " << endl;

cout << "4. Even/Odd" << endl;

cout << "What bet do you want to make of the following:" << endl;

cin >> plBtCh;

switch(plBtCh)

{

//Starting Roulette

case 1:

pritBet(chips, betChip, betsMad, vPlBtCh);

cout << " " << endl;

cout << "What number do you want to bet on:" << endl;

cin >> secChB;

spnRot(roll, black, red, column1, column2, column3, even, odd, green, color);

if(roll == secChB)

{

cout << " " << endl;

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

}

else

{

cout << " " << endl;

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

}

break;

//Column

case 2:

pritBet(chips, betChip, betsMad, vPlBtCh);

cout << " " << endl;

cout << "1. Column 1" << endl;

cout << "2. Column 2" << endl;

cout << "3. Column 3" << endl;

cout << "Which column do you want to bet on:" << endl;

cin >> secChB;

spnRot(roll, black, red, column1, column2, column3, even, odd, green, color);

switch(secChB)

{

//Starting Roulette

case 1:

for(int i = 0; i < 12; i++)

{

if(roll == column1[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2 + betChip;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

case 2:

for(int i = 0; i < 12; i++)

{

if(roll == column2[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2 + betChip;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

case 3:

for(int i = 0; i < 12; i++)

{

if(roll == column3[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2 + betChip;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

default:

cout << "You entered none of the above." << endl;

}

break;

//Red/Black

case 3:

pritBet(chips, betChip, betsMad, vPlBtCh);

cout << " " << endl;

cout << "1. Red" << endl;

cout << "2. Black" << endl;

cout << "Which color do you want to bet on:" << endl;

cin >> secChB;;

spnRot(roll, black, red, column1, column2, column3, even, odd, green, color);

switch(secChB)

{

//Starting Roulette

case 1:

for(int i = 0; i < 18; i++)

{

if(roll == red[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

case 2:

for(int i = 0; i < 18; i++)

{

if(roll == black[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

default:

cout << "You entered none of the above." << endl;

}

break;

case 4:

pritBet(chips, betChip, betsMad, vPlBtCh);

cout << " " << endl;

cout << "1. Even" << endl;

cout << "2. Odd" << endl;

cout << "Which number do you want to bet on:" << endl;

cin >> secChB;;

spnRot(roll, black, red, column1, column2, column3, even, odd, green, color);

switch(secChB)

{

//Starting Roulette

case 1:

for(int i = 0; i < 20; i++)

{

if(roll == even[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

case 2:

for(int i = 0; i < 18; i++)

{

if(roll == odd[i])

{

cout << "You won your bet" << endl;

chipWon = betChip \* 2;

chips += chipWon;

cout << "You won " << chipWon << " chips and your total chips are " << chips << endl;

break;

}

else

{

cout << "You lost your bet" << endl;

cout << "You lost " << betChip << " chips and your total chips are " << chips << endl;

break;

}

}

break;

default:

cout << "You entered none of the above." << endl;

}

break;

default:

cout << "You entered none of the above." << endl;

}

cout << "You have made a total of " << betsMad << " bet." << endl;

cout << "The best you have made so far are in order are:" << endl;

prntVec(vPlBtCh);

cout << " " << endl;

cout << "Would you like to play again? Yes/No: " << endl;

cin >> playAg;

cout << " " << endl;

// Checks if the first letter of the input of the user is Y/y. if yes do again.

}while(tolower(playAg[0]) == 'y');

cout << "Do you want to play a different game in the casino? Yes/No:" << endl;

cin >> playAg;

if(tolower(playAg[0]) == 'y')

{

menu(chips, betChip, chipWon, betsMad, vPlBtCh);

}

}

// Print the Roulette

void prntRot()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ " << endl;

cout << "| \_\_\_ | \_\_\_ \_\_\_ |" << endl;

cout << "| | | | | | | | |" << endl;

cout << "| | | | | | | | |" << endl;

cout << "| |\_\_\_| | |\_\_\_| |\_\_\_| |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_ " << endl;

cout << "| | | | |" << endl;

cout << "| 1 | 2 | 3 | \*LOW\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | 1 - 18 |" << endl;

cout << "| 4 | 5 | 6 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 7 | 8 | 9 | \*EVEN\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | |" << endl;

cout << "| 10 | 11 | 12 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 13 | 14 | 15 | \*RED\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | |" << endl;

cout << "| 16 | 17 | 18 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 19 | 20 | 21 | \*BLACK\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | |" << endl;

cout << "| 22 | 23 | 24 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 25 | 26 | 27 | \*ODD\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | |" << endl;

cout << "| 28 | 29 | 30 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 31 | 32 | 33 | \*HIGH\* |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_| |" << endl;

cout << "| | | | 19 - 36 |" << endl;

cout << "| 34 | 35 | 36 | |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

cout << "| | | | |" << endl;

cout << "| 2:1 | 2:1 | 2:1 |" << endl;

cout << "|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_|" << endl;

}

// Spin the Roulette

void spnRot(int roll, int black[], int red[], int column1[], int column2[],int column3[],int even[],int odd[], int green[], string color[])

{

roll = rand() % 37;

cout << roll << endl;

for(int i = 0; i < 20; i++)

{

if(roll == black[i])

{

for(int i = 0; i < 20; i++)

{

if(roll % 2 == 0)

{

cout << " " << endl;

cout << "The roulette rolled " << roll << color[0] << "even" << endl;

break;

}

if(roll % 2 == 1)

{

cout << " " << endl;

cout << "The roulette rolled " << roll << color[0] << "odd" << endl;

break;

}

}

}

if(roll == red[i])

{

for(int i = 0; i < 20; i++)

{

if(roll % 2 == 0)

{

cout << " " << endl;

cout << "The roulette rolled " << roll << color[1] << "even" << endl;

break;

}

if(roll % 2 == 1)

{

cout << " " << endl;

cout << "The roulette rolled " << roll << color[1] << "odd" << endl;

break;

}

}

}

if(roll == green[i])

{

if(roll == 37)

{

cout << " " << endl;

cout << "The roulette rolled " << "00" << color[2] << "even" << endl;

break;

}

if(roll == 0)

{

cout << " " << endl;

cout << "The roulette rolled " << roll << color[2] << "even" << endl;

break;

}

}

}

}

//Random numbers

void randNum(int chips, float betChip, int chipWon, int betsMad , vector<int> &vPlBtCh)

{

int size = 10; //Size of the array

int usRnNum[size]; //Create an array to store 10 user inputs

int value; //variable that creates the random numbers

int randNum[size]; // Array for the random numbers

string playAg;

bool machNum = false;

int ranAUI[size][size]; // 2d array for random Number and the user number

do

{

pritBet(chips, betChip, betsMad, vPlBtCh);

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

{

value = (rand()%10001);

randNum[i] = value ;

}

cout << "Enter 10 numbers between 10000 and 0"<<endl;

int usIn;

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

{

cin >> usIn;

while(usIn > 10000 || usIn < 0) // Doesn't let the user to input more chips than he has

{

cout << "This number is two big or small" << endl;

cout << "Enter a new number" << endl;

cin >> usIn;

}

usRnNum[i]= usIn;

}

cout << "This is the random number you got" << endl;

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

{

cout<<randNum[i]<<" ";

}

cout << " " << endl;

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

{

for(int j=0; j<size; j++)

{

if(usRnNum[i]==randNum[j])

{

machNum = true;

}

}

}

if(machNum == true)

{

cout << "You won your bet!" << endl;

chipWon = betChip \* 100;

chips =+ chipWon;

cout << "You bet was " << betChip << " and your total chips are " << chips << " chips." << endl;

}

else

{

cout << "You did not get any matching numbers."<<endl;

cout << "You lost " << betChip << " and you have left "<< chips << " chips." << endl;

}

cout << " " << endl;

// 2d Array

// I did not have a use for 2d array but here is how I used them

cout << "Printing the 2D array" << endl;

for (int row = 0; row < size; row++)

{

for (int cow = 0; cow < size; cow++)

{

ranAUI[row][cow] = usRnNum[row];

ranAUI[row][cow] = randNum[cow];

cout << "Element at x[" << row << "][" << cow << "]: ";

cout << ranAUI[row][cow]<<endl;

}

}

cout << " " << endl;

//Bubble sort

BublSor(randNum, size);

//Selection Sort

SelcSort(randNum, size);

//Binary Search

int userVal;

cout << "Enter an integer for the Binary Search: " << endl;

cin >> userVal;

int result = BinSrch(randNum, size, userVal);

if(result >= 0)

{

cout << "The number " << randNum[result] << " was found at the"

" element with index " << result << endl;

}

else

{

cout << "The number " << userVal << " was not found. " << endl;

}

cout << " " << endl;

cout << "You have made a total of " << betsMad << " bet." << endl;

cout << "The best you have made so far are in order are:" << endl;

prntVec(vPlBtCh);

cout << " " << endl;

cout << "Would you like to play again? Yes/No: " << endl;

cin >> playAg;

cout << " " << endl;

// Checks if the first letter of the input of the user is Y/y. if yes do again.

}while(tolower(playAg[0]) == 'y');

cout << "Do you want to play a different game in the casino? Yes/No:" << endl;

cin >> playAg;

if(tolower(playAg[0]) == 'y')

{

menu(chips, betChip, chipWon, betsMad, vPlBtCh);

}

}

**5. Proof of a working product**

z