PROJECT 2

Elias Roulette Table Game

Elias Silva

Course CSC 5

Section 46332

7/24/20222

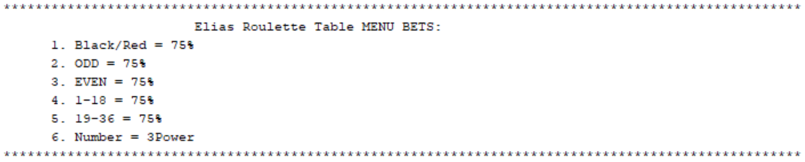
Description:

Player will be able to gamble on a roulette table. They will be six bet options where, the player, gets to select one. The player has the option to bet any amount possible. They will be informed on the winning returns if the bet is successful. There are six bet options:

1. Bet of Spin is Black or Red
2. Bet of the Spin number is ODD
3. Bet of the Spin number is EVEN
4. Bet of the Spin number is between 1-18
5. Bet of the Spin number is between 19-36
6. Bet of the Spin number is guessed correct.

If player wins bet numbers 1 through 5 -player will win 75% on the initial bet. If player hits the jackpot bet number 6-player will win the third power of the initial bet.

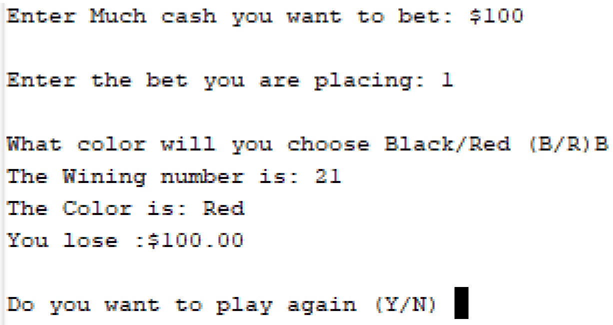
Menu:



```
****************************************************************************************************
                        Elias Roulette Table MENU BETS:
      1. Black/Red = 75%
      2. ODD = 75%
      3. EVEN = 75%
      4. 1-18 = 75%
      5. 19-36 = 75%
      6. Number = 3Power
****************************************************************************************************
```

Player will be display with this menu option be inputting the amount of money to be bet. After the input of money, the player will select which bet to play.

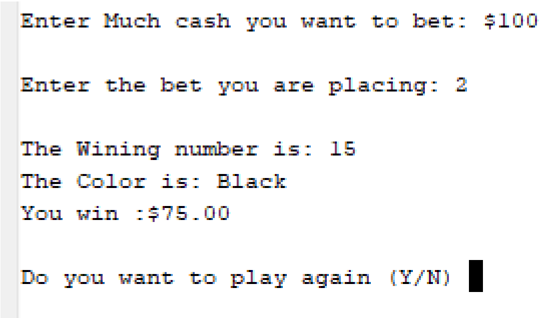
Bet 1:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 1

What color will you choose Black/Red (B/R)B
The Wining number is: 21
The Color is: Red
You lose :$100.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 1. Player will input an option of Black or Red. Displays the wining color, wining number, and the won or lost in the game.

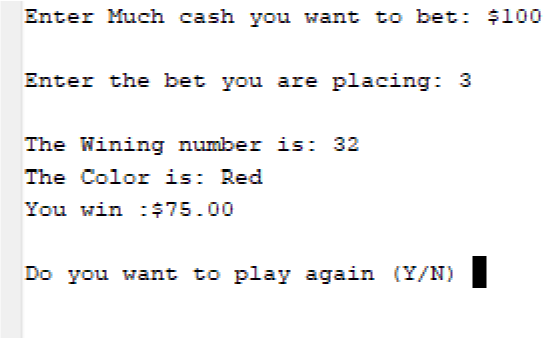
Bet 2:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 2

The Wining number is: 15
The Color is: Black
You win :$75.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 2. Displays the wining color, wining number, and the won or lost in the game. (The wining number was an ODD).

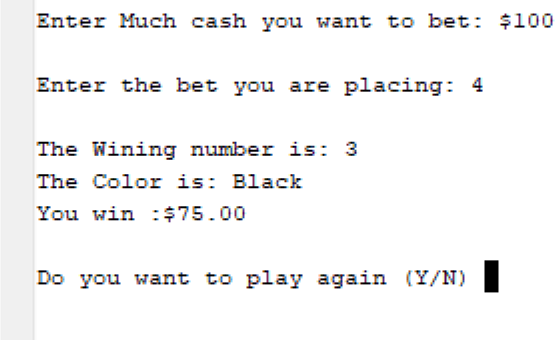
Bet 3:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 3

The Wining number is: 32
The Color is: Red
You win :$75.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 3. Displays the wining color, wining number, and the won or lost in the game. (The wining number was an EVEN).

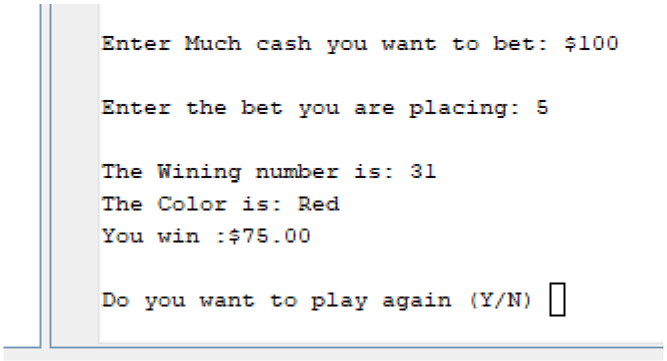
Bet 4:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 4

The Wining number is: 3
The Color is: Black
You win :$75.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 4. Displays the wining color, wining number, and the won or lost in the game.(The wining number was is between 1-18.

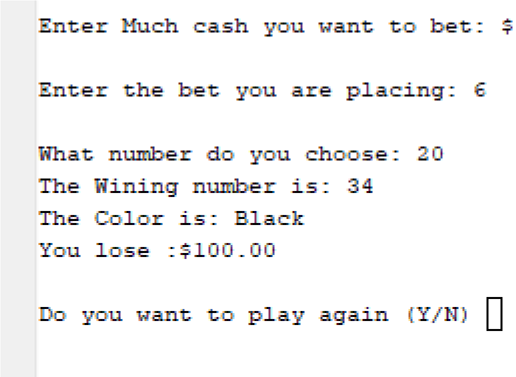
Bet 5:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 5

The Wining number is: 31
The Color is: Red
You win :$75.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 5. Displays the wining color, wining number, and the won or lost in the game. (The wining number was is between 18-36).

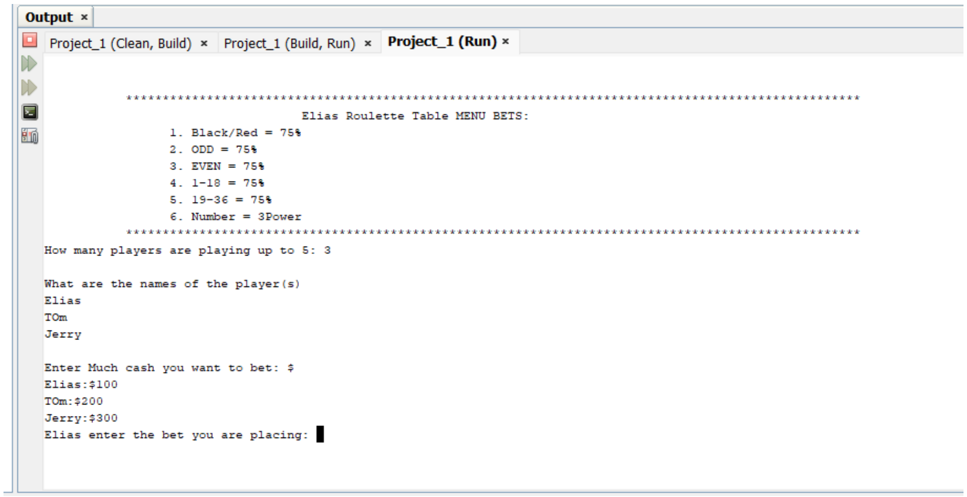
Bet 6:



```
Enter Much cash you want to bet: $100

Enter the bet you are placing: 6

What number do you choose: 20
The Wining number is: 34
The Color is: Black
You lose :$100.00

Do you want to play again (Y/N)
```

Player will input the amount of money and bet 6. Displays the wining color, wining number, and the won or lost in the game. (The wining number was 34, guess was 20).

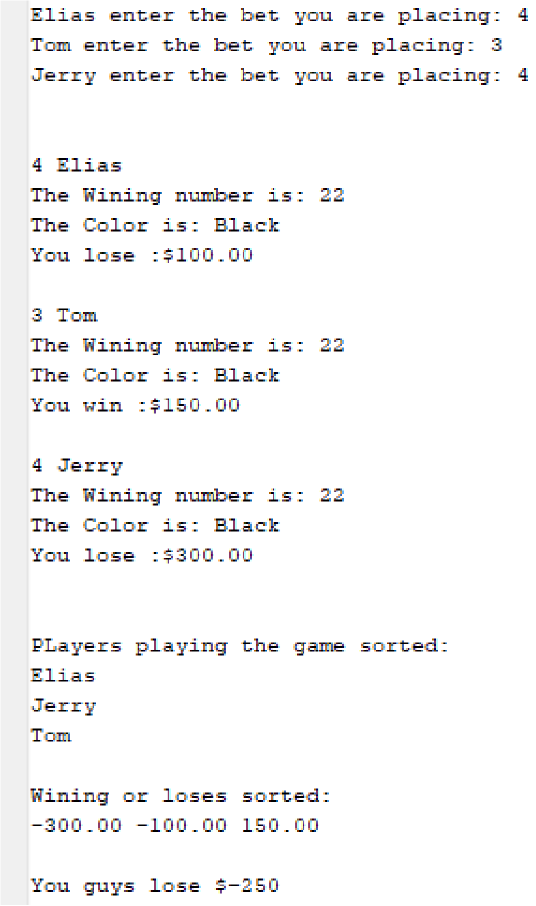
Inputs



```
           ****************************************************************************************************
                                   Elias Roulette Table MENU BETS:
                 1. Black/Red = 75%
                 2. ODD = 75%
                 3. EVEN = 75%
                 4. 1-18 = 75%
                 5. 19-36 = 75%
                 6. Number = 3Power
           ****************************************************************************************************
How many players are playing up to 5: 3

What are the names of the player(s)
Elias
TOm
Jerry

Enter Much cash you want to bet: $
Elias:$100
TOm:$200
Jerry:$300
Elias enter the bet you are placing: █
```

Ask the user for the number of players and the names of the players. The users will select the amount of money they would like to bet and the bet based of the menu.

Display

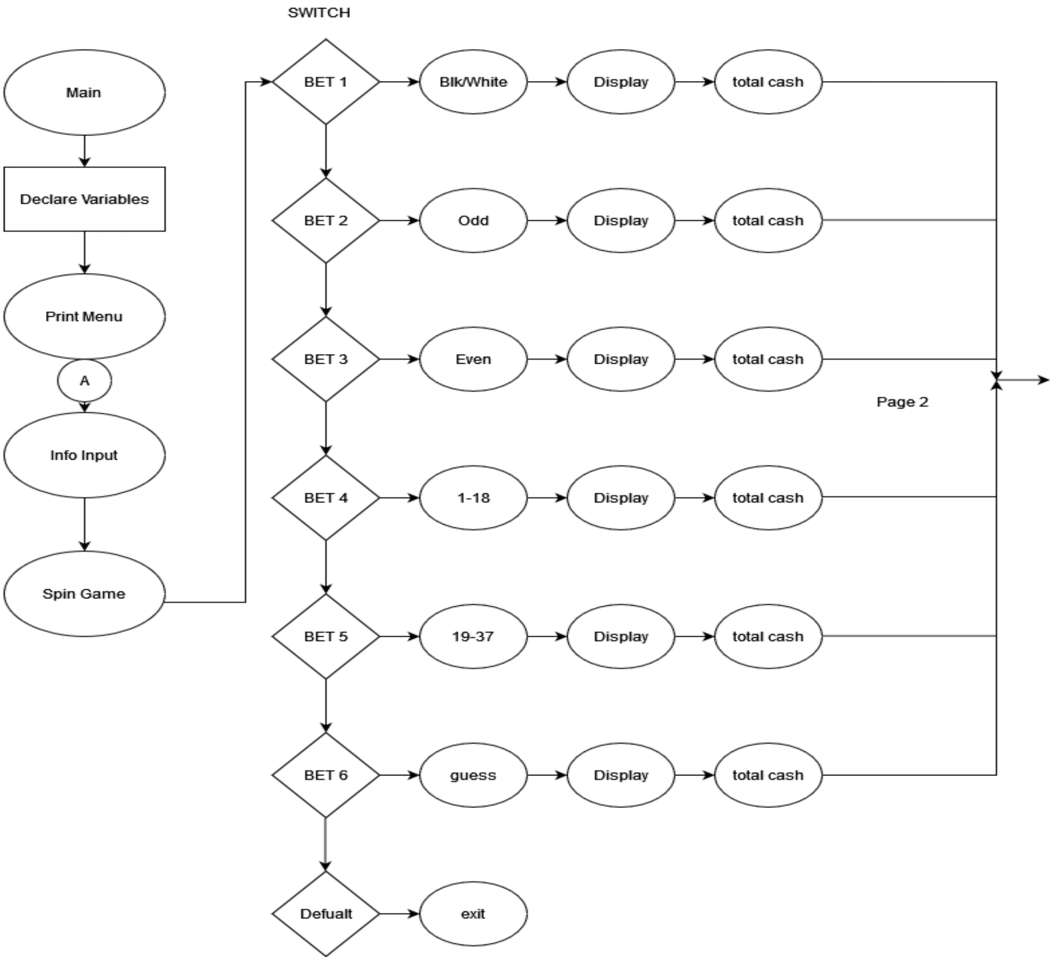


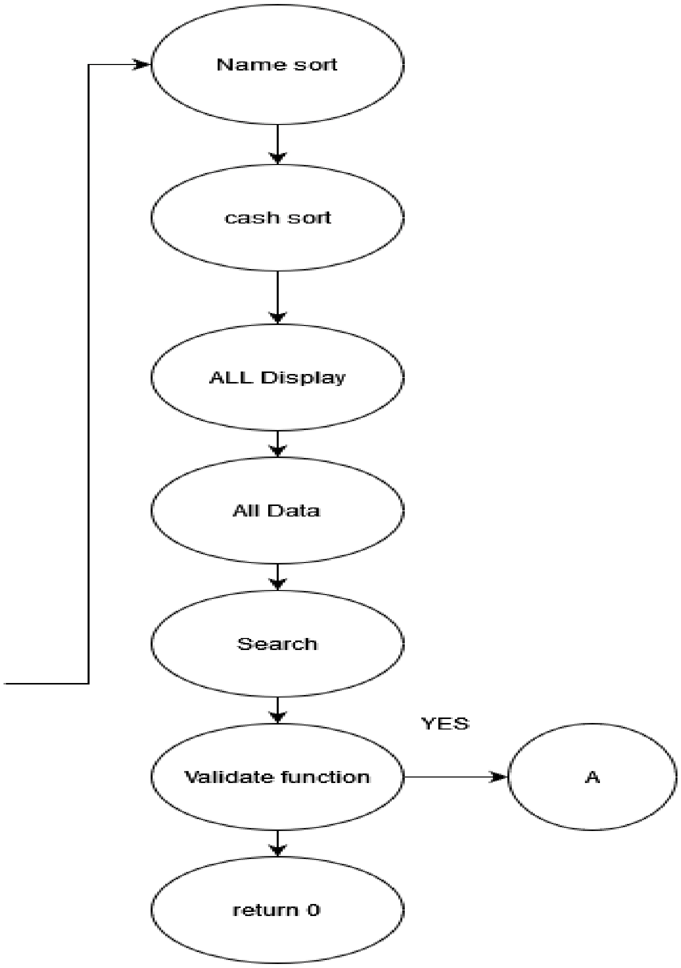
```
Elias enter the bet you are placing: 4
Tom enter the bet you are placing: 3
Jerry enter the bet you are placing: 4


4 Elias
The Wining number is: 22
The Color is: Black
You lose :$100.00

3 Tom
The Wining number is: 22
The Color is: Black
You win :$150.00

4 Jerry
The Wining number is: 22
The Color is: Black
You lose :$300.00


PLayers playing the game sorted:
Elias
Jerry
Tom

Wining or loses sorted:
-300.00 -100.00 150.00

You guys lose $-250
```

The game will display the individual game result per user bet. The game will result a sorted ascending of the loses to wins per group. The game will calculate the winnings or loses as group.

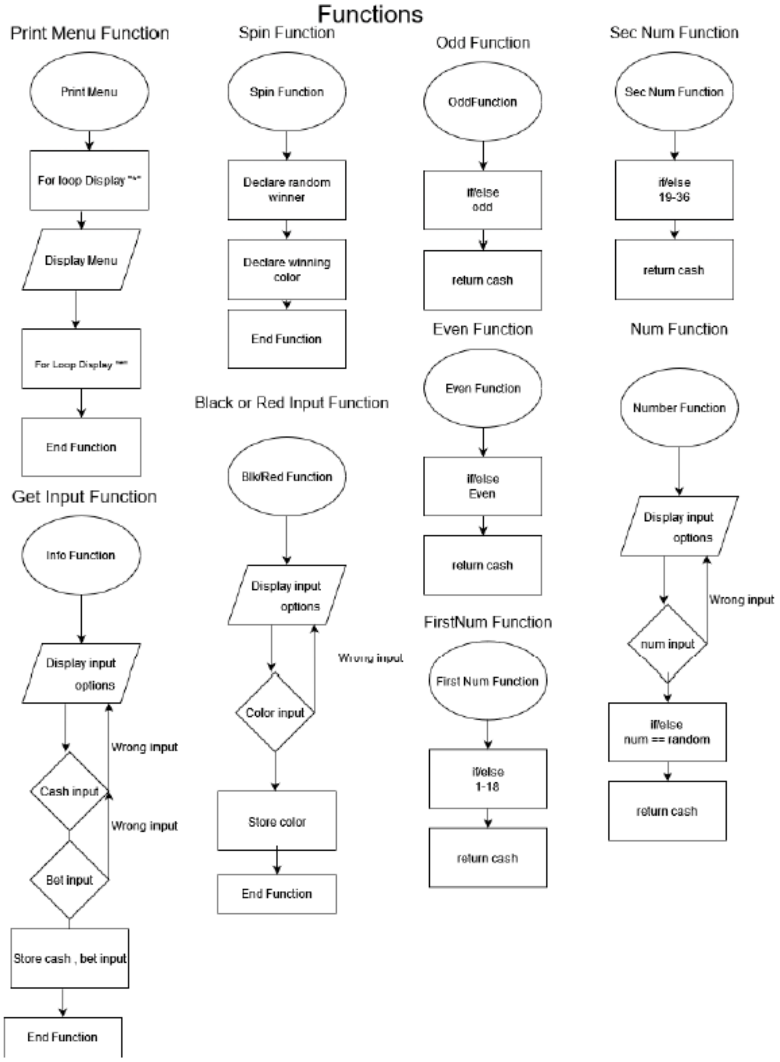
Cross Reference: **ON THE PDF FILE**

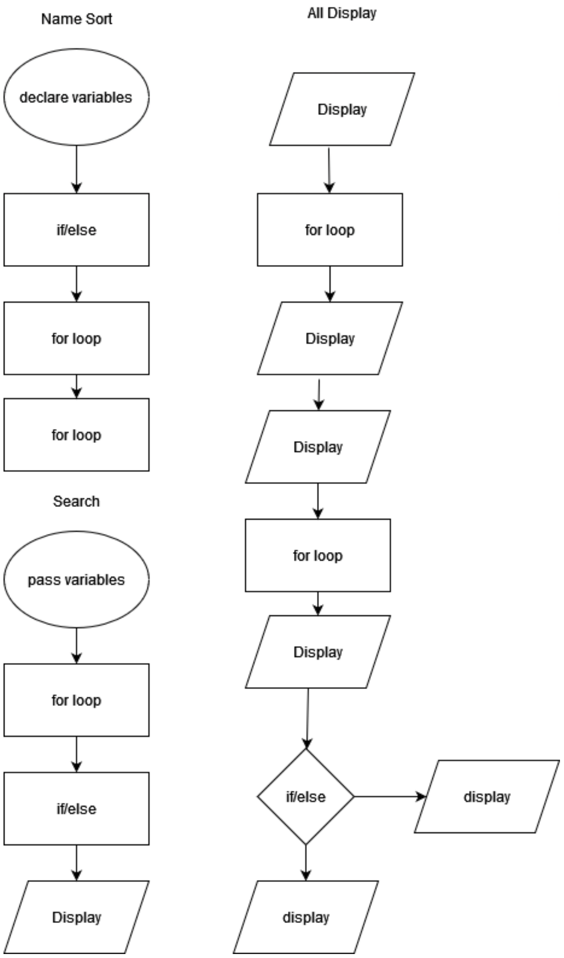
Flow Chart:



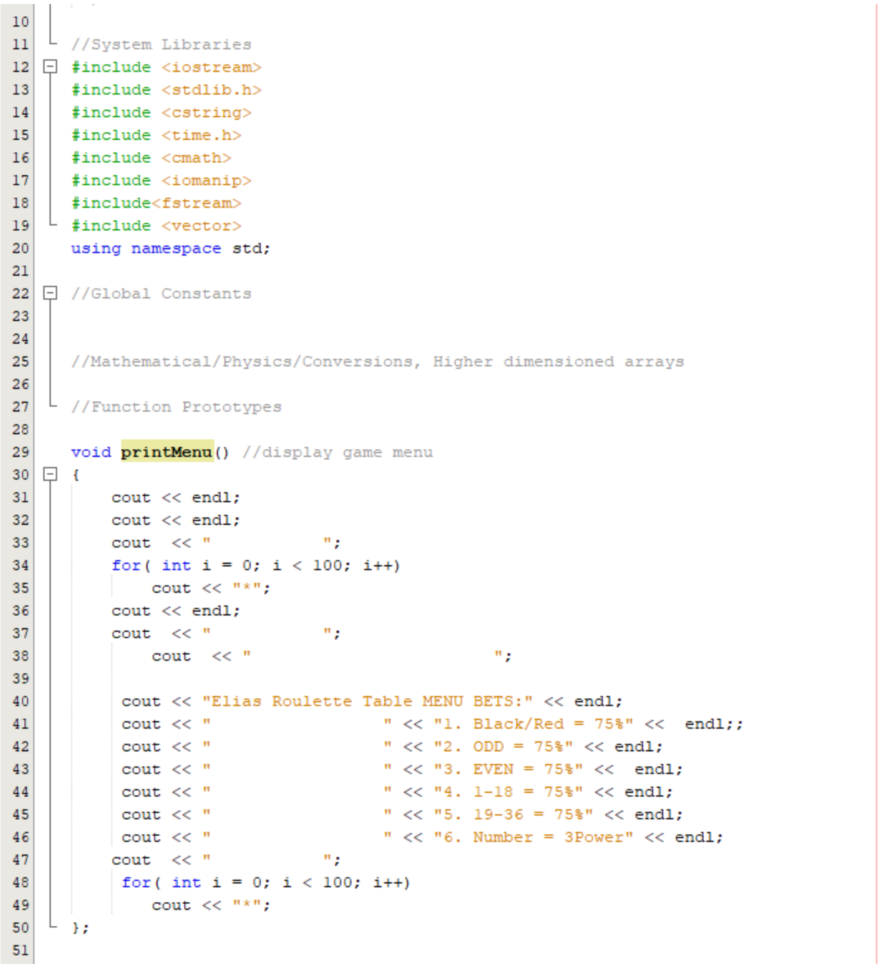

Name sort
cash sort
ALL Display
All Data
Search
Validate function
YES
A
return 0

Functions

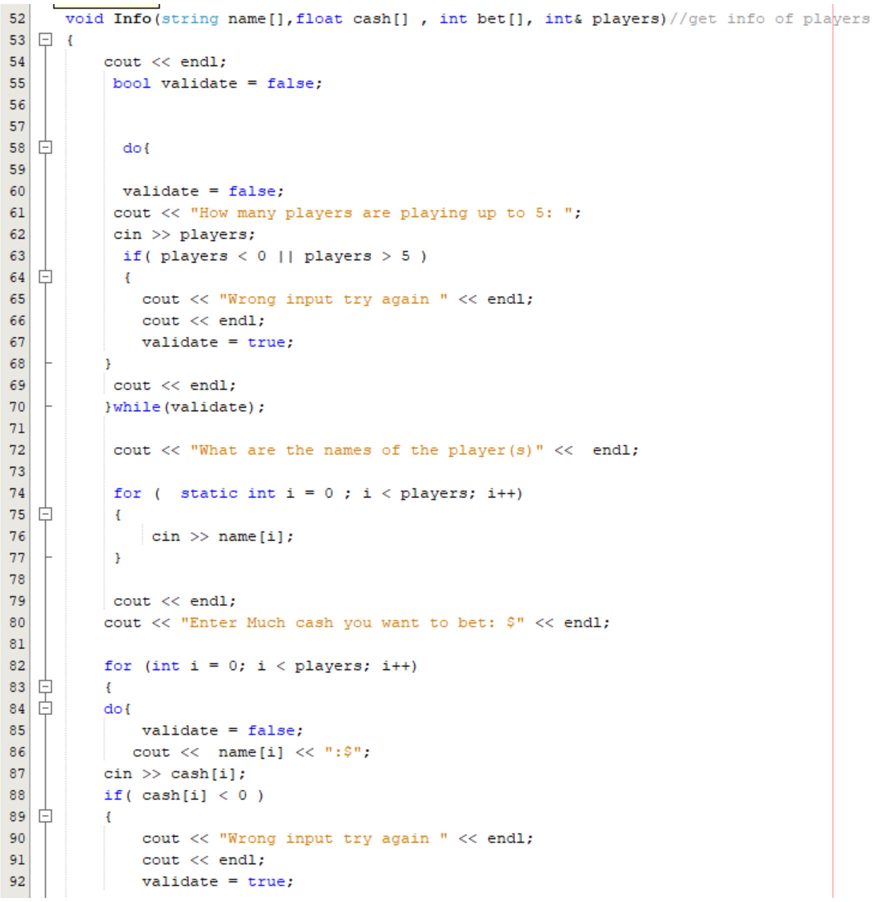

Print Menu Function
Print Menu
For loop Display "*"
Display Menu
For Loop Display "*"
End Function
Get Input Function
Info Function
Display input options
Wrong input
Cash input
Wrong input
Bet input
Store cash , bet input
End Function
Spin Function
Spin Function
Declare random winner
Declare winning color
End Function
Black or Red Input Function
Blk/Red Function
Display input options
Wrong input
Color input
Store color
End Function
Odd Function
OddFunction
if/else odd
return cash
Even Function
Even Function
if/else Even
return cash
FirstNum Function
First Num Function
if/else 1-18
return cash
Sec Num Function
Sec Num Function
if/else 19-36
return cash
Num Function
Number Function
Display input options
Wrong input
num input
if/else num == random
return cash


Name Sort
declare variables
if/else
for loop
for loop
Search
pass variables
for loop
if/else
Display
All Display
Display
for loop
Display
Display
for loop
Display
if/else
display
display

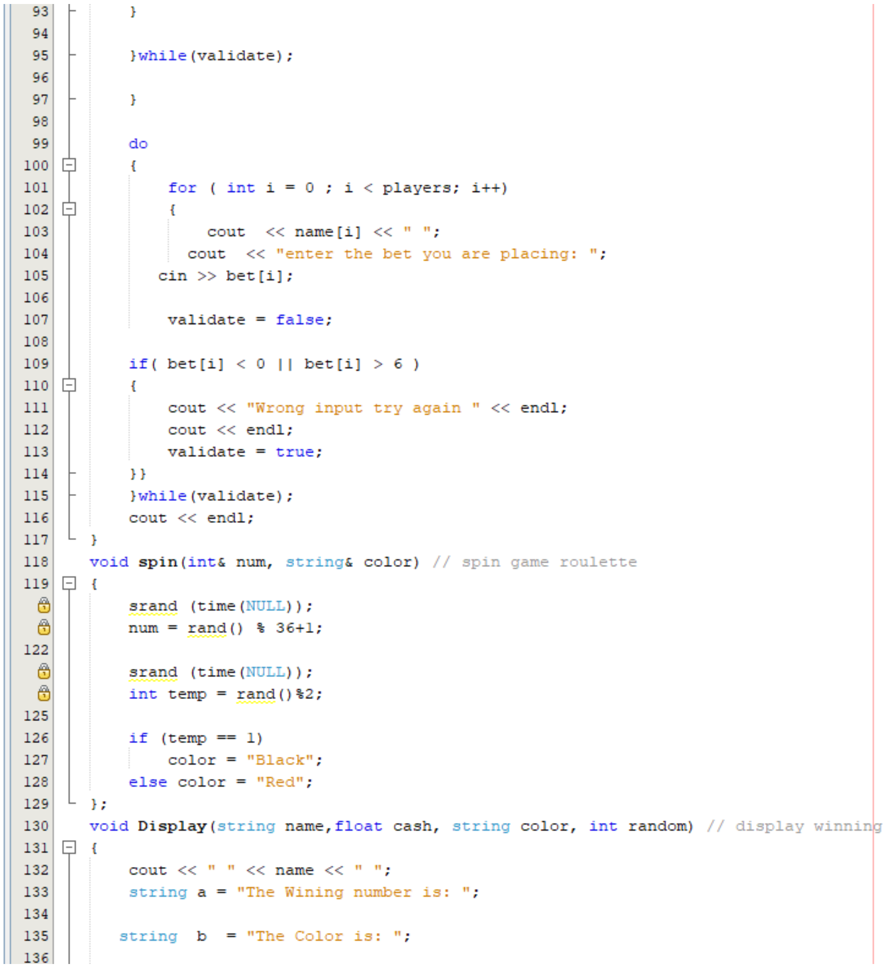
Code:



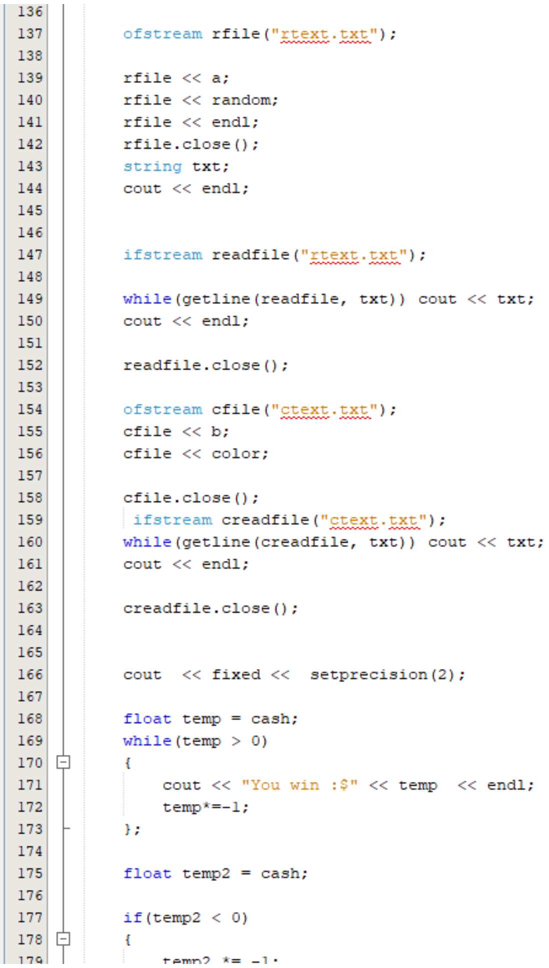
```
//System Libraries
#include <iostream>
#include <stdlib.h>
#include <cstring>
#include <time.h>
#include <cmath>
#include <iomanip>
#include<fstream>
#include <vector>
using namespace std;

//Global Constants


//Mathematical/Physics/Conversions, Higher dimensioned arrays

//Function Prototypes

void printMenu() //display game menu
{
    cout << endl;
    cout << endl;
    cout  << "           ";
    for( int i = 0; i < 100; i++)
        cout << "*";
    cout << endl;
    cout  << "           ";
        cout  << "                        ";

     cout << "Elias Roulette Table MENU BETS:" << endl;
     cout << "                 " << "1. Black/Red = 75%" <<  endl;;
     cout << "                 " << "2. ODD = 75%" << endl;
     cout << "                 " << "3. EVEN = 75%" <<  endl;
     cout << "                 " << "4. 1-18 = 75%" << endl;
     cout << "                 " << "5. 19-36 = 75%" << endl;
     cout << "                 " << "6. Number = 3Power" << endl;
    cout  << "           ";
     for( int i = 0; i < 100; i++)
        cout << "*";
};
```



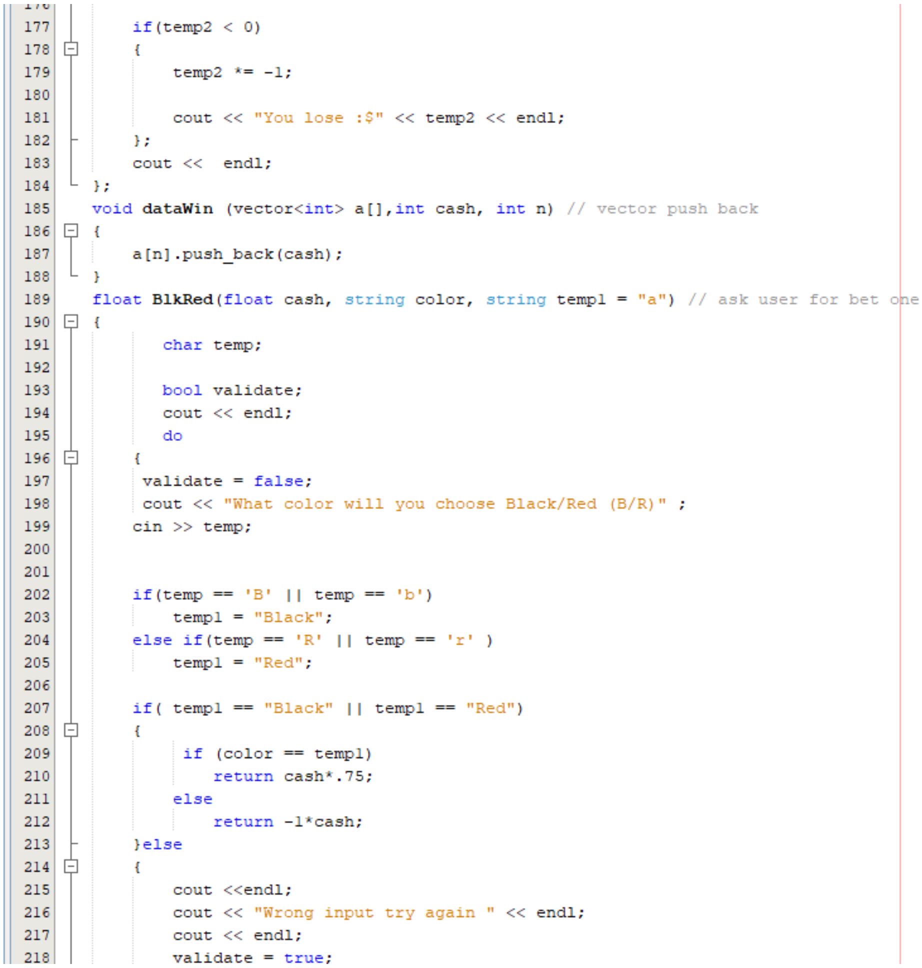
```
void Info(string name[],float cash[] , int bet[], int& players)//get info of players
{
    cout << endl;
     bool validate = false;


      do{

      validate = false;
     cout << "How many players are playing up to 5: ";
     cin >> players;
      if( players < 0 || players > 5 )
      {
        cout << "Wrong input try again " << endl;
        cout << endl;
        validate = true;
    }
     cout << endl;
    }while(validate);

     cout << "What are the names of the player(s)" <<  endl;

     for (  static int i = 0 ; i < players; i++)
     {
         cin >> name[i];
     }

     cout << endl;
    cout << "Enter Much cash you want to bet: $" << endl;

    for (int i = 0; i < players; i++)
    {
    do{
        validate = false;
       cout <<  name[i] << ":$";
    cin >> cash[i];
    if( cash[i] < 0 )
    {
        cout << "Wrong input try again " << endl;
        cout << endl;
        validate = true;
```



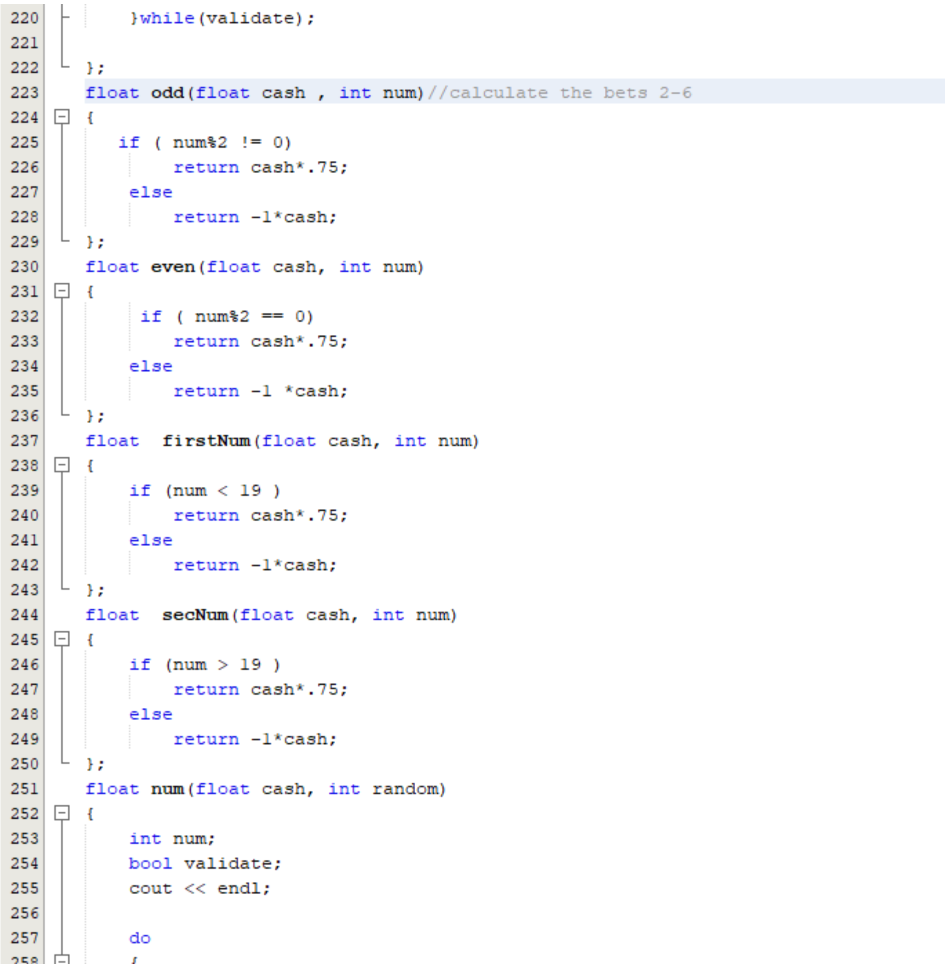
```
    }

    }while(validate);

    }

    do
    {
        for ( int i = 0 ; i < players; i++)
        {
            cout  << name[i] << " ";
          cout  << "enter the bet you are placing: ";
       cin >> bet[i];

        validate = false;

    if( bet[i] < 0 || bet[i] > 6 )
    {
        cout << "Wrong input try again " << endl;
        cout << endl;
        validate = true;
    }}
    }while(validate);
    cout << endl;
}
void spin(int& num, string& color) // spin game roulette
{
    srand (time(NULL));
    num = rand() % 36+1;

    srand (time(NULL));
    int temp = rand()%2;

    if (temp == 1)
        color = "Black";
    else color = "Red";
};
void Display(string name,float cash, string color, int random) // display winning
{
    cout << " " << name << " ";
    string a = "The Wining number is: ";

   string  b  = "The Color is: ";

```



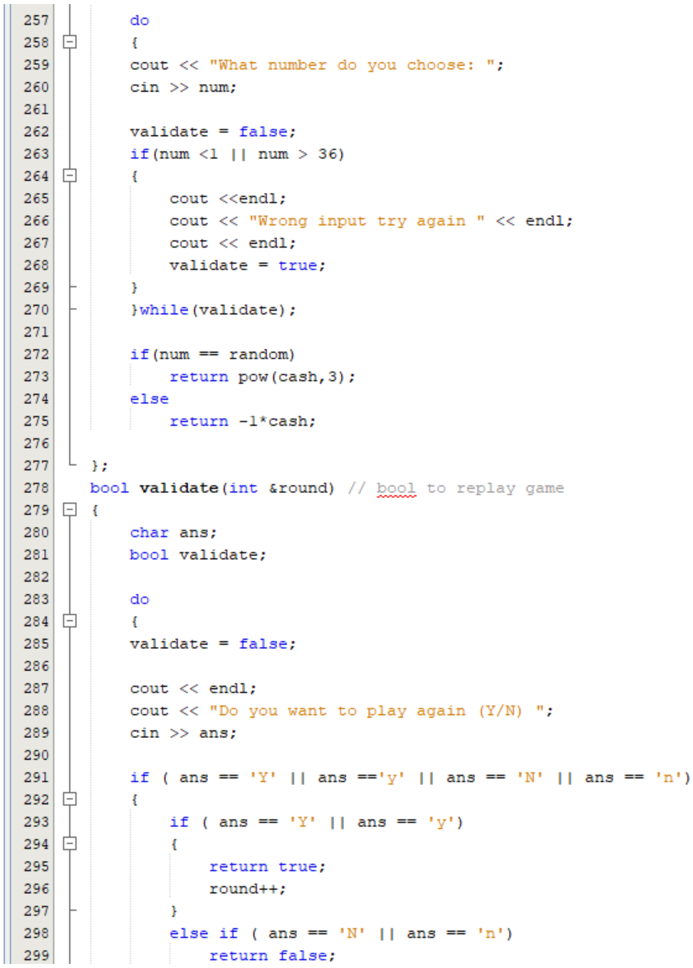
```
ofstream rfile("rtext.txt");

rfile << a;
rfile << random;
rfile << endl;
rfile.close();
string txt;
cout << endl;


ifstream readfile("rtext.txt");

while(getline(readfile, txt)) cout << txt;
cout << endl;

readfile.close();

ofstream cfile("ctext.txt");
cfile << b;
cfile << color;

cfile.close();
 ifstream creadfile("ctext.txt");
while(getline(creadfile, txt)) cout << txt;
cout << endl;

creadfile.close();


cout  << fixed <<  setprecision(2);

float temp = cash;
while(temp > 0)
{
    cout << "You win :$" << temp  << endl;
    temp*=-1;
};

float temp2 = cash;

if(temp2 < 0)
{
    temp2 *= -1;
```



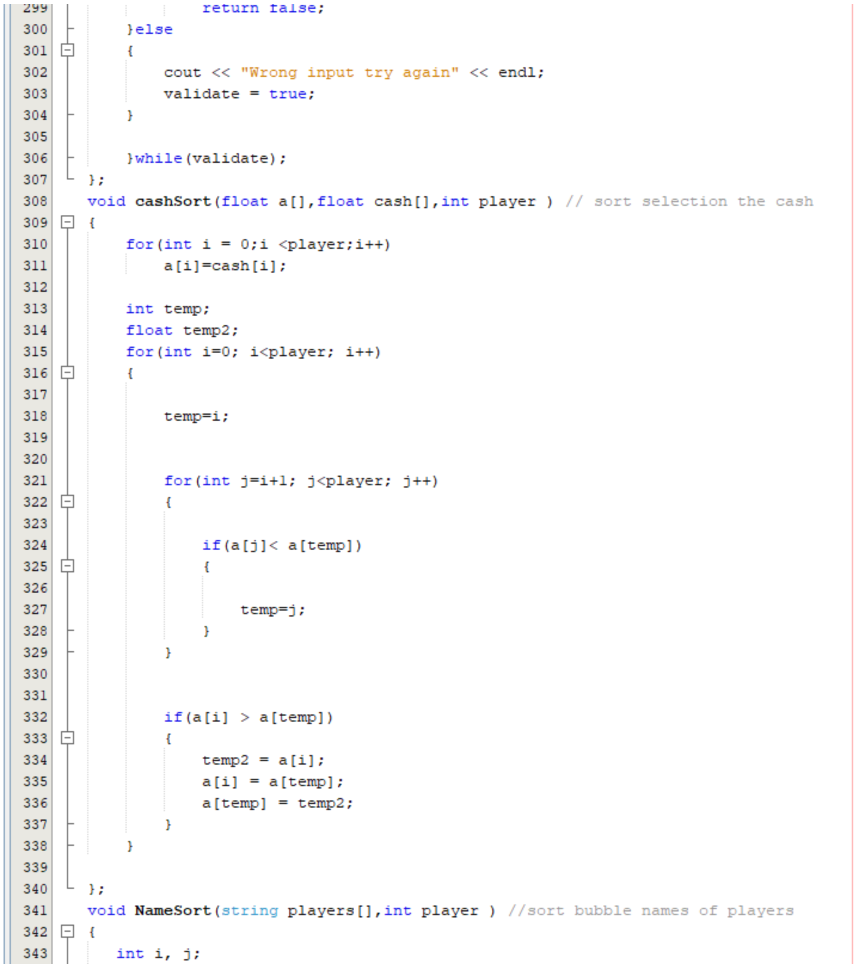
```
    if(temp2 < 0)
    {
        temp2 *= -1;

        cout << "You lose :$" << temp2 << endl;
    };
    cout <<  endl;
};
void dataWin (vector<int> a[],int cash, int n) // vector push back
{
    a[n].push_back(cash);
}
float BlkRed(float cash, string color, string templ = "a") // ask user for bet one
{
       char temp;

       bool validate;
       cout << endl;
       do
    {
     validate = false;
     cout << "What color will you choose Black/Red (B/R)" ;
    cin >> temp;


    if(temp == 'B' || temp == 'b')
        templ = "Black";
    else if(temp == 'R' || temp == 'r' )
        templ = "Red";

    if( templ == "Black" || templ == "Red")
    {
         if (color == templ)
            return cash*.75;
        else
            return -1*cash;
    }else
    {
        cout <<endl;
        cout << "Wrong input try again " << endl;
        cout << endl;
        validate = true;
```



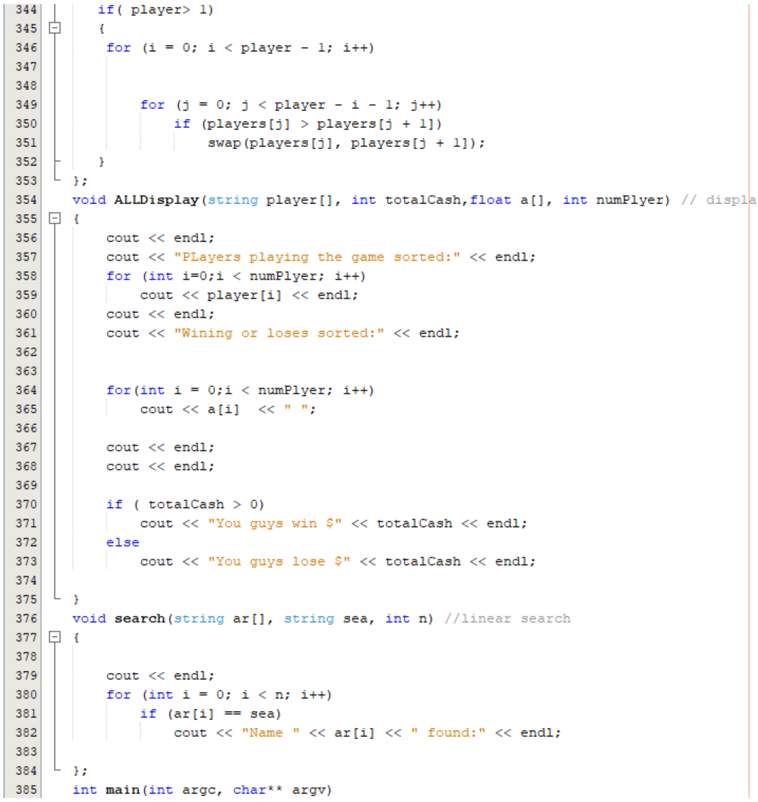
```
    }while(validate);

};
float odd(float cash , int num)//calculate the bets 2-6
{
   if ( num%2 != 0)
        return cash*.75;
    else
        return -1*cash;
};
float even(float cash, int num)
{
     if ( num%2 == 0)
        return cash*.75;
    else
        return -1 *cash;
};
float  firstNum(float cash, int num)
{
    if (num < 19 )
        return cash*.75;
    else
        return -1*cash;
};
float  secNum(float cash, int num)
{
    if (num > 19 )
        return cash*.75;
    else
        return -1*cash;
};
float num(float cash, int random)
{
    int num;
    bool validate;
    cout << endl;

    do
    {
```



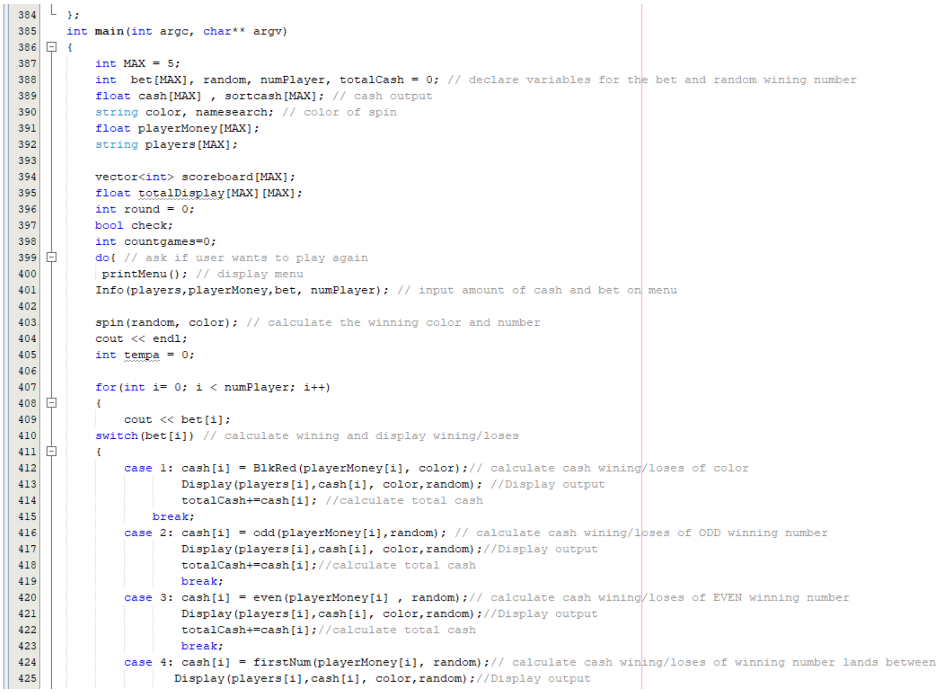
```
    do
    {
    cout << "What number do you choose: ";
    cin >> num;

    validate = false;
    if(num <1 || num > 36)
    {
        cout <<endl;
        cout << "Wrong input try again " << endl;
        cout << endl;
        validate = true;
    }
    }while(validate);

    if(num == random)
        return pow(cash,3);
    else
        return -1*cash;

};
bool validate(int &round) // bool to replay game
{
    char ans;
    bool validate;

    do
    {
    validate = false;

    cout << endl;
    cout << "Do you want to play again (Y/N) ";
    cin >> ans;

    if ( ans == 'Y' || ans =='y' || ans == 'N' || ans == 'n')
    {
        if ( ans == 'Y' || ans == 'y')
        {
            return true;
            round++;
        }
        else if ( ans == 'N' || ans == 'n')
            return false;
```



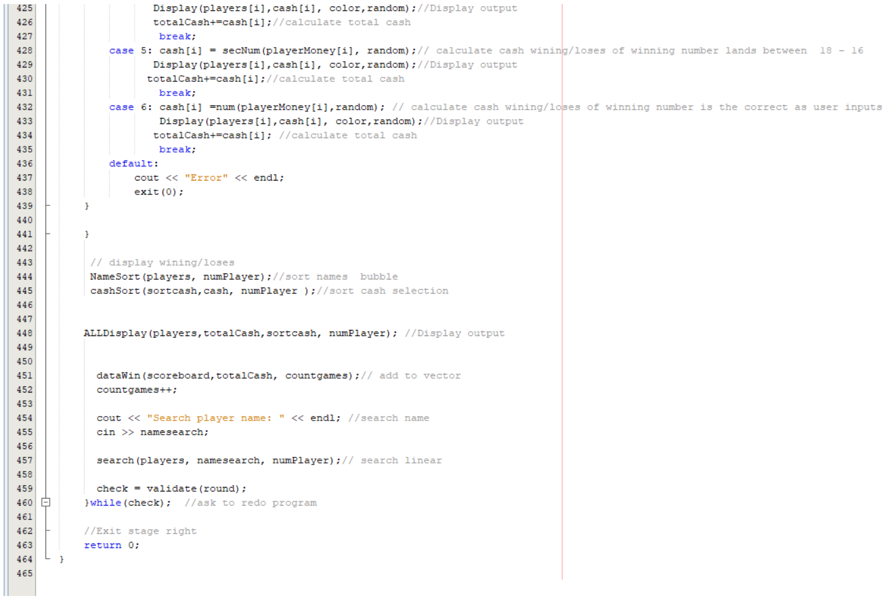
```
            return false;
    }else
    {
        cout << "Wrong input try again" << endl;
        validate = true;
    }

    }while(validate);
};
void cashSort(float a[],float cash[],int player ) // sort selection the cash
{
    for(int i = 0;i <player;i++)
        a[i]=cash[i];

    int temp;
    float temp2;
    for(int i=0; i<player; i++)
    {

        temp=i;


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

            if(a[j]< a[temp])
            {

                temp=j;
            }
        }


        if(a[i] > a[temp])
        {
            temp2 = a[i];
            a[i] = a[temp];
            a[temp] = temp2;
        }
    }

};
void NameSort(string players[],int player ) //sort bubble names of players
{
   int i, j;
```



```cpp
   if( player> 1)
   {
    for (i = 0; i < player - 1; i++)


        for (j = 0; j < player - i - 1; j++)
            if (players[j] > players[j + 1])
                swap(players[j], players[j + 1]);
   }
};
void ALLDisplay(string player[], int totalCash,float a[], int numPlyer) // display all out put
{
    cout << endl;
    cout << "PLayers playing the game sorted:" << endl;
    for (int i=0;i < numPlyer; i++)
        cout << player[i] << endl;
    cout << endl;
    cout << "Wining or loses sorted:" << endl;


    for(int i = 0;i < numPlyer; i++)
        cout << a[i]  << " ";

    cout << endl;
    cout << endl;

    if ( totalCash > 0)
        cout << "You guys win $" << totalCash << endl;
    else
        cout << "You guys lose $" << totalCash << endl;

}
void search(string ar[], string sea, int n) //linear search
{

    cout << endl;
    for (int i = 0; i < n; i++)
        if (ar[i] == sea)
            cout << "Name " << ar[i] << " found:" << endl;

};
int main(int argc, char** argv)
```



```cpp
};
int main(int argc, char** argv)
{
    int MAX = 5;
    int  bet[MAX], random, numPlayer, totalCash = 0; // declare variables for the bet and random wining number
    float cash[MAX] , sortcash[MAX]; // cash output
    string color, namesearch; // color of spin
    float playerMoney[MAX];
    string players[MAX];

    vector<int> scoreboard[MAX];
    float totalDisplay[MAX][MAX];
    int round = 0;
    bool check;
    int countgames=0;
    do{ // ask if user wants to play again
     printMenu(); // display menu
    Info(players,playerMoney,bet, numPlayer); // input amount of cash and bet on menu

    spin(random, color); // calculate the winning color and number
    cout << endl;
    int tempa = 0;

    for(int i= 0; i < numPlayer; i++)
    {
        cout << bet[i];
    switch(bet[i]) // calculate wining and display wining/loses
    {
        case 1: cash[i] = BlkRed(playerMoney[i], color);// calculate cash wining/loses of color
                Display(players[i],cash[i], color,random); //Display output
                totalCash+=cash[i]; //calculate total cash
            break;
        case 2: cash[i] = odd(playerMoney[i],random); // calculate cash wining/loses of ODD winning number
                Display(players[i],cash[i], color,random);//Display output
                totalCash+=cash[i];//calculate total cash
                break;
        case 3: cash[i] = even(playerMoney[i] , random);// calculate cash wining/loses of EVEN winning number
                Display(players[i],cash[i], color,random);//Display output
                totalCash+=cash[i];//calculate total cash
                break;
        case 4: cash[i] = firstNum(playerMoney[i], random);// calculate cash wining/loses of winning number lands between
               Display(players[i],cash[i], color,random);//Display output
```



```
               Display(players[i],cash[i], color,random);//Display output
               totalCash+=cash[i];//calculate total cash
                break;
        case 5: cash[i] = secNum(playerMoney[i], random);// calculate cash wining/loses of winning number lands between  18 - 16
               Display(players[i],cash[i], color,random);//Display output
              totalCash+=cash[i];//calculate total cash
                break;
        case 6: cash[i] =num(playerMoney[i],random); // calculate cash wining/loses of winning number is the correct as user inputs
                Display(players[i],cash[i], color,random);//Display output
               totalCash+=cash[i]; //calculate total cash
                break;
        default:
            cout << "Error" << endl;
            exit(0);
    }

    }

     // display wining/loses
     NameSort(players, numPlayer);//sort names  bubble
     cashSort(sortcash,cash, numPlayer );//sort cash selection


    ALLDisplay(players,totalCash,sortcash, numPlayer); //Display output


      dataWin(scoreboard,totalCash, countgames);// add to vector
      countgames++;

      cout << "Search player name: " << endl; //search name
      cin >> namesearch;

      search(players, namesearch, numPlayer);// search linear

      check = validate(round);
    }while(check);  //ask to redo program

    //Exit stage right
    return 0;
}
```