

# PROJECT 2

*<SLOT MACHINE v2>*

CSC5 - 40107

Cay, Shienne Patricia

Date: 02/11/2017

# Introduction

Title: Slot Machine v2

Slot Machine is played just as how people play it in casinos. Once a pattern comes out, the player wins the bet. Amount of money won varies by bet amount and the pattern it won with. This is the main idea of the game.

However, I added some features so I could utilize topics in class that were not used version 1. The game starts as soon as it is debugged. The program will display the winning patterns and prompt the user for a decision. Before, it was only to play or to cancel. But now, user can choose to play a mini-game within the program. It's a number guessing game with a twist. User don't necessarily guess the number but takes a chance if input number occurs 5 times in printing, user will then earn \$10 even with no cash insertion. However, one wouldn't be able to get the money from mini-game if player does not play the main game. Mini game winnings are added to cash bonus.

Another feature I added is the data feature that is only accessible to casino "staffs". I set the game up to only accommodate 25 play count and if the number of maximum players is reached, the program will prompt for a Staff Login and reset the data. Even though it can be reset in the program, saved file of user's game report can still be retrieved as it is in an external file. Inside the staff access are the players game information like the total cash involved, cash loss, cash win and cash bonus. Another thing is, the only way for the program to stop is when the "staff" resets the game.

For the player list, if a user comes back and plays again, user will have the option to look at his/her last game report before it is overwritten by new game play.

**Note:** Just like how it goes in slot machines in any casinos, bet amount is first deducted to the current money before it generates the random numbers. Once the player wins, the bet amount is added to current money, and is multiplied depending on the pattern won.

Example: If user has \$50 and bets \$25 dollars, and it wins a 2x bet pattern. Though user won \$50 dollars, only \$25 dollars will be added to his/her account. The money that the user will have on his/her account will be \$75.

It is a game won purely by luck.

## How It Works

Once user hit play, the program will show the main menu.

Listed there are the winning patterns and jackpot. Number of players who played the game is in there as well as the current money which works when user choose to play mini-game. The winnings from mini-game will be added there. However, if user chooses to play game and cancel it after, user will still be added to player list for game record since player would have to insert cash after typing name before the prompt for cancel is given.

```
CSCS CASINO RIVERSIDE

SLOT MACHINE 2017 (C++) Version

Winning Patterns:                                JACKPOT:

X X X  - - -  - - -                                X - X  - - -  X X X
- - -  X X X  - - -                                - X -  7 7 7  X - X
- - -  - - -  X X X                                X - X  - - -  X X X

X - -  - - x                                - X -  X X X
- X -  - X -                                X X X  X X X
- - X  x - -                                - X -  X X X

Players Listed: 0
Current Money: 0

Would you like to play the game?
Y - YES      N - NO
M - MINI GAME

Note: Casino Staff access S for settings.

CHOICE: 
```

There are four choices in main menu which user can access.

Y – It will prompt user to enter the game with his/her name and insert cash. This program also recognizes if user is a new player or not. If a user is not a new player, program will give player a choice to see previous game report or not before proceeding to inserting cash.

```
CHOICE: Y

Enter Player Name: User

You are a new player. Welcome User!

Insert Cash: $100

Thank you for choosing to play the game!
We added $1.00 to your capital as cash bonus.

Good luck ^_^

-----
Player: User
Current Money: $101.01
-----
```

```
CHOICE: N

Goodbye!

Play the game next time!
```

N – This will tell you user to play The game next time and it will go back to main menu. Another thing is, if user played the mini-game and won but proceeded not to play the slot machine, hitting 'N' will reset the current money back to 0.

M – This will activate the Mini-game which doesn't require a name and cash insert. Winning in mini-game will earn \$5 and no downside if user loses. The player must play the slot machine before mini-game winning is added to the recorded money and bonus. Otherwise, it will remain in menu until someone plays for it or hits 'N' for reset. Playing mini-game will not add user to player list.

```
CHOICE: M

WELCOME TO THE NUMBER GUESS MINI-GAME!

Rules: You must input a number and if it exists in the array for
3 times, you win $5 to be added to your current money. You will
be able to cash it out if you proceed to play the slot machine
after the mini-game. If you don't, the money will remain unless
you choose not to play the game.

PLAY?      P - PLAY      C - CANCEL
```

S – This will direct user for staff login to display database/information of the players who played the slot machine. Everyone who inserted cash is listed in the list.

```
CHOICE: S

Staff Login

Employee Name: username
Input Password: password
```

**Note:** To access this option, the Employee Name is “username” and the password is “password” just like in the picture. Entering a different one will show log in failure and will go back to main menu.

```
PLAY

S - SPIN   C - CANCEL

CHOICE: S

BET: $10

   8  4  8
   6  7  8
   8  4  6

Sorry, you lost $10.00

-----
Player: User
Current Money: $91.01
-----
```

For the actual play of the game, it only requires an input for Spin or Cancel, and a bet if use decides to Spin. The program will then generate random numbers for machine and it will tell if loser won or not. It will loop again until user chooses to cancel.

If user chooses to cancel the game and still have money left, the program will print out a voucher for cashout. If user used up all his/her money, no voucher will be printed the instance he/she cancels the game and not insert more cash.

```
CHOICE: C

Cashout voucher printing...

-----
CSCS CASINO RIVERSIDE, CA                                3679

CASHOUT VOUCHER

FOR User

Validation  80-3975-8546-7232-3677
Sat Feb 11 14:32:14 2017

$91.01

Ticket void after 30 days      Machine #1062204352
-----

NOTE: Bring your ID with you to cash it out to verify
name in the voucher.

Would you like to see your game report?
Y - YES      N - NO

CHOICE: █
```

## Additional Photos/Screenshots

WELCOME TO THE NUMBER GUESS MINI-GAME!

Rules: You must input a number and if it exists in the array for 3 times, you win \$5 to be added to your current money. You will be able to cash it out if you proceed to play the slot machine after the mini-game. If you don't, the money will remain unless you choose not to play the game.

PLAY? P - PLAY C - CANCEL

CHOICE: P

Input a number from 10-99!

Number: 15

```
10 10 10 11 12 12 12 13 13 13 13 14 14 15 16 16 17 17 17 18
19 20 21 22 23 24 24 24 25 25 25 25 25 26 26 26 26 27 27 28
28 29 29 30 30 30 30 31 31 31 32 33 34 34 35 35 36 36 36 36
37 37 37 37 37 38 38 38 38 39 40 40 41 42 43 44 44 44 45 46
46 47 47 48 48 48 49 49 50 50 51 52 52 52 53 54 54 54 55 56
56 56 56 56 56 57 58 58 59 59 59 60 60 60 60 61 62 63 63 65
65 66 66 68 69 69 69 70 70 71 71 71 71 72 72 73 73 73 73 73
74 74 74 74 74 75 75 75 76 76 76 76 77 77 78 78 78 78 79
80 81 81 82 83 83 84 84 86 87 87 87 87 87 87 87 88 88 89 89
90 90 90 90 90 91 91 93 93 93 93 93 94 95 95 95 97 98 99 99
```

Number 15 occurred 1 times.

Sorry, you didn't win. Try again!

PLAY? P - PLAY C - CANCEL

CHOICE: P

Input a number from 10-99!

Number: 23

```
10 10 11 12 12 12 12 13 13 13 13 14 14 15 16 17 18 19 19 19
20 20 20 21 22 22 23 23 23 23 23 23 24 24 24 25 25 25 26 26
26 26 26 26 27 28 29 29 30 30 30 30 30 30 33 33 33 33 34 34 35
35 36 36 37 37 38 40 41 41 42 42 43 43 44 44 45 46 46 47 47
48 48 48 48 49 49 49 50 51 51 52 52 52 53 53 55 55 55 55 55
56 56 56 57 57 58 58 58 59 59 59 59 59 60 60 61 62 63 64 64
65 66 66 66 67 68 68 69 69 69 69 70 71 71 72 72 73 73 74 74
75 76 76 76 77 77 78 78 79 79 81 81 81 82 83 83 83 83 83 84
85 86 86 87 87 87 87 88 88 89 89 89 89 89 90 91 91 92 92 92
93 94 94 95 95 95 95 96 96 97 97 97 97 98 99 99 99 99 99 99
```

Number 23 occurred 6 times.

Congratulations! \$5 is added to the current money!

Staff Login

Employee Name: anything  
Input Password: anything

Accessing database...

Log in failed!

Warning: Enter only either 'S' or 'C'!

Warning: You can only enter 'Y' or 'N'!

Player: User

Current Money: \$0.00

You have used up all your funds. Would you like to insert again?

Y - Yes N - No

CHOICE: Y

Insert Cash: \$1500

Player: User

Current Money: \$1500.00

Enter Player Name: User

Welcome back, User!

Would you like to see your previous game report?

Y - YES N - NO

CHOICE: █

Warning: You can only enter 'Y' or 'N' or 'M'!

Congratulations!

You earned game bonus for reaching 10 number of spins!  
\$54.5 is added to your cash amount!

BET: \$25

Your bet cannot be higher than your capital!  
Your remaining cash is \$2.01



[illegible]

CHOICE: ☐

7	6	5
4	5	6
5	8	8

```
Player: User1
Current Money: $24.51
```

7	6	4
7	7	7
6	6	5

Player List	Total Spins	Total Cash	Cash Bonus	Cash Won	Cash Loss
User1	9	1272.21	115.61	0.61	1271.22
User2	3	1012.17	43.17	312.00	246.00

CHOICE: ☐

## Summary

Project Size: 965 lines

The Number of Variables: about 60

I have applied every topic required to show for the cross reference list even the input fstream that I was not able to include in the version 1. It took me almost a week to complete version 2 without errors as much as possible. I only added one part to the project that we have not learned yet which is outputting the current time and date as the program ends and debugs. I find it necessary to add that for the cash out voucher since I would not be able to create a barcode. The voucher looks a bit like a check then because of it.

I did not change much from the version 1 but only applied newly learned concepts from class.

Other than that, the game is still full of nested do-while loops and switch. I had a hard time formatting the 2D array which is paralleled to a string array and is also paralleled to a vector. For some reasons, `<iomanip>` did not work the first time. I also had to adjust a lot of values especially passing values and making a copy because I encountered playing the game with the money from before remain intact even after the previous player is done. It was a bit rigorous but it works now.

## Description

The main point I programmed in this project is formatting the 2D array paralleled to a string array and vector to make a table of information for staff access. I also had to make changes to make the mini-game possible.



## Cross Reference for Projects 1

Chapter	Section	Topic	Where in code – Line Number
<b>2</b>	2	Cout	104-110, 134-137, 160-162, 455-460 and more.
	3	Libraries	iostream, iomanip, fstream, cstdlib, cmath, string, ctime, vector
	4	Variables/Literals	51-98, 468, 573
	5	Identifiers	51-98, 468, 573
	6	Integers	76-83, 468, 573
	7	Characters	84-90
	8	Strings	54-58
	9	Floats no doubles	64-74
	10	Bools	94-98
	11	Sizeof *****	n/a
	12	Variables 7 characters or less	checked
	13	Scope ***** no global variables	n/a
	14	Arithmetic Operators	checked – 148-149, 156-157...
	15	Comments 20% +	checked
	16	Named Constants	51-52
	17	Programming Style *****	n/a
<b>3</b>	1	Cin	111, 127, 138, 175, 187, 407, 420, 464
	2	Math Expression	checked – 148-149, 156-157
	3	Mixing data types *****	n/a
	4	Overflow/Underflow *****	n/a
	5	Type Casting	698-699
	6	Multiple Assignment *****	n/a
	7	Formatting Output	570
	8	Strings	54-58
	9	Math Library	151
	10	Hand Tracing *****	n/a
<b>4</b>	1	Relational Operators	checked – 151-154...
	2	If	118, 155, 424...
	4	If-else	420-421, 386-392...
	5	Nesting	177-140
	6	If-else-if	545-551, 470-496...
	7	Flags *****	n/a
	8	Logical Operators	checked – 217, 369...
	11	Validating User Input	182-187, 398, 441...
	13	Conditional Operator	151-154

	14	Switch	122, 409
<b>5</b>	1	Increment/Decrement	199, 223, 395... play++, numWin++, numLose++
	2	While	182, 475, 915
	5	Do-while	102, 117, 169, 171, 178, 403, 412, 461
	6	For loop	584
	11	Files input/output	Input – 128-131    Output – 687-703
	12	No breaks in loops *****	checked
<b>6</b>	3	Prototypes	29-43
	4	Pass by Value	29, 33
	8	Returning Values	898, 931
	9	Boolean return values	884-885
	11	Static Local Variable	883, 896, 910
	12	Default Arguments	32
	13	Reference Variable	29
	14	Overloading functions	30-31
<b>7</b>	4	Array Initialization	56, 75, 81
	7	Parallel Arrays	534-541
	8	Arrays in function arguments	33, 35-37
	9	2 Dimensional Arrays	75
		/as function argument	--
	12	STL Vector	53
<b>8</b>	1	Search Linear/Binary	484
	3	Sorting Bubble/Selection	483
	5	Applied to Vectors	--

## Pseudo-code

```
/*
 * File:  main.cpp
 * Author: Shienne Cay
 * Created on February 8, 2017, 7:49 PM
 * Purpose: PROJECT 2 SLOT MACHINE
 */

//System Libraries

//User Libraries

//Global Constants
//Such as PI, Vc, -> Math/Science values
//as well as conversions from one system of measurements
//to another

//Percentage Conversion
//Two Dimensional Arrays

//Function Prototypes
//Function to reset all variables for use to 0 after one cancels game
//View game report after a game
//View game report from previous play
//Bonus cash calculation from number of spins reached
//Search a number
//Display voucher
//Fill array with random number elements
//Sorting array for easier win clarification
//Print the array or the random numbers in mini-game
//Name and current money
//Password validation for administrator settings
//Round off Budget
//Insert budget/money
//Display Winning Patterns
//Prompt for game report

//Executable code begins here! Always begins in Main
```

```
//Set the random seed using time
```

```
//Declare Variables
```

```
//Input Values
```

```
//Display Winning patterns
```

```
//Display times played count
```

```
//Money so far added from mini-game
```

```
//Prompt user to play game or not
```

```
//If Yes, start loop.
```

```
//If players count is greater than size for array, prompt administrator for reset
```

```
//If user chooses to play the game
```

```
//Prompt user for name to be used later for cashout voucher using string datatype
```

```
//If user played before, retrieve play count from game report saved on file
```

```
//Prompt returning user if he/she wants to see previous game report
```

```
//If yes, retrieve whole game report and display
```

```
//Upon displaying, reset all retrieved data back to 0
```

```
//Add player's name to players list
```

```
//If user is a new player, add player's name to list
```

```
//Welcome user
```

```
//Prompt for cash deposit/insert
```

```
//Add the amount inserted to total cash inserted but subtract money from mini-  
game
```

```
//Give cash bonus if Player decided to play
```

```
    //If inserted amount is below 1000, get 1% of cash and multiply by itself
```

```
    //If inserted amount is >1000 but <=2000, get 10% of cash
```

```
    //If over 2000, get 5% of cash and add to the cash of user
```

```
//Round off Cash Bonus to two decimal places to get exact value
```

```
//Tell the user that the system added cash bonus to account
```

```
//Add the cash bonus to overall cash bonus in record
```

```
//Call rndOffB function, status function
```

```
//Start loop for budget condition. If bool x for budget is true, continue loop
```

```
//If budget is greater than 0, start loop.
```

```
//Prompt user to Spin or Cancel game
```

```
//Any other input will not be accepted
```

```
//If user chose to spin and budget is greater than 0
```

```
    //Prompt user for betting amount
```

```
    //User input validation of float datatype for bet amount
```

```
        //If bet is greater than the budget, display invalidity
```

```

        //If bet is equal to 0, display invalidity
        //If bet is less than or equal to budget and budget is greater than 0
            //Increment total number of spins
            //Generate slot machine numbers
            //Determine if user won
                //If user won, add corresponding price to budget,
                increment won and add win to total cash win
                //If user lost, subtract bet from budget, increment loss
                and add loss to total loss
            //Determine if gameplay reached
            //If reached add cash bonus, increment gameplay, add bonus to total cash
bonus
        //Call rndOffB function, status function
        //Loop for spin or cancel if budget>0, bet is less than 0 and budget is > 0.

        //If user chose cancel at the start
            //If total spin count is 0 and user chose not to play game after money
            inserted
            //Deduct cash bonus from inserted amount
            //Deduct cash bonus added for total record
            //Exit loop for spin
        //If spin count is >1, exit loop

        //If user input is not 'S' or 'C', display invalidity, ask again
        //Continue loop til budget is 0, choice is not to spin and boolean for boolean
for budget remains true

        //If budget is 0 or turns 0, or if cash is used up
        //Start loop for cash insert
        //Prompt user if players wants to insert more cash to continue playing
            //If yes, ask how much
                //Start loop for add amount
                //User input validation of float datatype for adding funds
                //If add amount is is greater than 0, add to budget or current money
                //If add amount is <= 0, diplay invalidity
            //Amount inserted is added into total cash inserted record
            //Continue loop if added inserted cash input is less than 0
        //Boolean for add game turns false. Exit loop for add cash

        //If choice is 'n' or 'N'
            //Boolean for add game turns false.. exit loop for add cash
            //Boolean for budget turns false, exit game play loop

```

```

//If user input is not 'Y' or 'N', display invalidity
//Continue loop if bool z for add cash remains true

//If no conditions are met exit game play loop

//Continue loop if bool x for budget remains true or budget is greater than 0,
continue loop
//Exit game play

//If user chose Mini game, welcome user and explain rules
    //Prompt user to play or cancel/exit mini game
    //Start to fill array
    //Prompt user to input number
        //User input validation of float datatype for number
    //Sort the filled array
    //Find out how many times user input's number appeared
    //Print sorted array
    //Display result
        //If user input occurred 3 times or more, add $5 to money
            //Add cash won from mini-game to total cash bonus
        //If user didn't win, money remains 0
        //Reset found count
    //If user chooses to cancel game, exit mini-game and back to main menu
    //Else, warn user to input right character. Goes back to main menu
//Continue mini-game as long as user chooses to play
//Exit mini-game back to main menu

//If begin is 'n' or 'N', exit game prompt
    //Set money to 0 if player did not choose to play even if player won in mini
game
    //Game play exit, back to main menu

//If user enters staff login access
    //Prompt user for username and password to log in and access player's
important records
        //If logged in, display game player's information
        //Prompt for reset after viewing records
        //If chose to reset, end game
            //If not, back to main menu
            //If user input is not Y or N, repeat process
        //If employee name and password is wrong, display login failed. Go back to
main menu

```



```

//If begin is not 'N' or 'Y' or 'M' or 'S', display invalidity

//If game play remains true, continue loop

//Start loop exit condition
//Loop count for 2d array
//Calculate total money involved
//Add important information to database for staff settings
//Increase row
//Decrease column back to 0
//Increment total player count
//Loop voucher one time
//If chosen to exit game and there's money left on account
//Display voucher
//Prompt user for game report
//If yes, display report – function call
//Save report to file
//Reset all data and back to main menu
//If user does not want to see game report
//Save report to file
//Reset all data and back to main menu
//If user input is not either 'Y' or 'N', display invalidity
//If no money left on account
//Prompt user to see game report
//If yes, display report
//Save report to file
//If user does not want to see game report
//Save report to file
//Reset all data and back to main menu
//If user input is not either 'Y' or 'N', display invalidity
//Once condition reached, end game.

//If user decides not to play game, back to main menu and reset the bearned money from
mini-game if there is.

```

# Program

```
/*
 * File:  main.cpp
 * Author: Shienne Cay
 * Created on January 27, 2017, 3:57 PM
 * Purpose: PROJECT 1 SLOT MACHINE
 * Problem:
 */

#include <iostream>
#include <string>
#include <cstdlib>
#include <ctime>
#include <iomanip>
#include <cmath>
#include <fstream>
using namespace std;

const short PERCENT = 100;
const int ROWS = 25, COLS = 4;

void resetA(int &, int &, int &, float &, float &, float &, float &, float &, float &, float &, float &, float&);
void gameRep(string, char, int, int, int, float, float, float, float, float, float, float);
void gameRep(string, int, int, int, float, float, float, float, float, float, float);
float bonCash(float, float, float, float, float, float, int, int, float = 0);
void find(int [], int, int &, int);
void voucher(string, float);
void filArray(int [], int);
void mrkSrt(int [], int);
void prntAry(int [], int);
void status(float, string);
bool valPass(string, string);
float rndOffB(float, int);
float insert(float &, string);
void winPat();
void askRep();

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

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

    const int SIZE = 25;
    const int MINSIZE = 200;
    vector<int> spins(SIZE);
    string name,
            dummy;
    const float bonCsh1 = 0.05f,
            bonCsh2 = 0.10f,
```

```

        bonCsh3 = 0.15f,
        bonCsh4 = 0.25f,
        bonCshP = 0.01f,
        minVal = 0;
float budget = 0,
    add,
    bet,
    loss = 0,
    win = 0,
    bonWin = 0,
    minCsh = 0,
    cshIns = 0,
    totCash,
    winPer,
    lossPer,
    data[ROWS][COLS] = { };
int numWin = 0,
    numLose = 0,
    play = 0,
    g8mBon = 10,
    rndOff = 100,
    array[MINSIZE] = { },
    found = 0;
int count = 0, plCount = 0;
char begin,
    load,
    choice,
    report
    start,
    game,
    reset;
unsigned short rn1, rn2, rn3,
    rn4, rn5, rn6,
    rn7, rn8, rn9;
bool w = true;
bool x = true;
bool y = true;
bool z = true;
bool rep = true;

do {
    winPat();
    cout<<"\nPlayers Listed: "<<plCount<<endl;
    cout<<"Current Money: "<<budget<<endl;
    cout<<"\nWould you like to play the game? "<<endl;
    cout<<"    Y - YES    N - NO"<<endl;
    cout<<"    M - MINI GAME"<<endl<<endl;
    cout<<"Note: Casino Staff access S for settings."<<endl<<endl;
    cout<<"CHOICE: ";
    cin>>begin;
    getline(cin, dummy);

```

```

rep=true;
x=true;
z=true;

do {
    if (plCount>=SIZE) {
        cout<<"\nMaximum number of players reached. \nAdministrator must login!"<<endl;
        begin = 'S';
    }
    switch (begin) {
        case 'y':
        case 'Y': {
            cout<<endl;
            cout<<"Enter Player Name: ";
            getline(cin, name);
            ifstream in;
            in.open(name+".dat");
            in>>play;
            in.close();

            if (play>0) {
                cout<<"\nWelcome back, "<<name<<"!"<<endl<<endl;
                cout<<"Would you like to see your previous game report?"<<endl;
                cout<<"    Y - YES    N - NO"<<endl<<endl;
                cout<<"CHOICE: ";
                cin>>report;
                if (report == 'Y' || report == 'y')
                    gameRep(name,play,numWin,numLose,cshIns,win,bonWin,totCash,loss,winPer,lossPer);
                resetA(play, numWin, numLose, cshIns, win, bonWin, totCash, loss, winPer, lossPer,
                    budget);
                names[count] = name;
            }
            else if (play<=0){
                names[count] = name;
                cout<<"\nYou are a new player. Welcome "<<name<<"!"<<endl;
            }

            budget = insert(budget, dummy);
            cshIns += (budget-minCsh);

            float cshBon = (budget<=1000)?pow((budget*bonCshP), 2):
                (budget>1000&&budget<=2000)?budget*bonCsh2:
                (budget>=2000)?budget*bonCsh1:minVal;

            if (budget >= 0 && budget <= 20000) {
                float exCshBn = cshBon*rndOff+0.5;
                cshBon = exCshBn/100.0f;

                budget += cshBon;
                cout<<"\nThank you for choosing to play the game!"<<endl;
                cout<<"We added $"<<cshBon<<" to your capital as cash bonus."<<endl<<endl;
            }
        }
    }
} while (begin != 'S');

```

```

        cout<<"Good luck ^_^"<<endl;
        bonWin+=cshBon;
    }

    budget = rndOffB(budget, rndOff);
    status(budget, name);

    do {
        if (budget > minVal) {
            do {
                cout<<"\tPLAY"<<endl<<endl;
                cout<<"S - SPIN  C - CANCEL"<<endl<<endl;
                cout<<"CHOICE: ";
                cin>>choice;

                if ((choice == 'S' || choice == 's') && budget>minVal)
                    do {
                        cout<<endl<<"BET: $";
                        cin>>bet;

                        while (cin.fail()) {
                            cout<<"\nYou must only enter a number!"<<endl<<endl;
                            cout<<"Bet: $";
                            cin.clear();
                            getline(cin, dummy);
                            cin>>bet;
                        }

                        if (bet>budget) {
                            cout<<"\nYour bet cannot be higher than your capital!"<<endl;
                            cout<<"Your remaining cash is $"<<budget<<endl<<endl;
                        }
                        else if (bet<=minVal) {
                            cout<<"\nBet must be greater than 0!"<<endl;
                            cout<<"Your remaining cash is $"<<budget<<endl<<endl;
                        }
                        else if (bet<=budget && bet>minVal) {
                            play++;

                            rn1 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn2 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn3 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn4 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn5 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn6 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn7 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn8 = (rand()%(8-4+1))+4; //Value range from 4 to 8
                            rn9 = (rand()%(8-4+1))+4; //Value range from 4 to 8

                            cout<<endl;
                            cout<<"\t"<<rn1<<" "<<rn2<<" "<<rn3<<endl;

```

```

cout<<"\t"<<rn4<<" "<<rn5<<" "<<rn6<<endl;
cout<<"\t"<<rn7<<" "<<rn8<<" "<<rn9<<endl<<endl;

if (rn1==rn2 && rn2==rn3 && rn3==rn4 && rn4 == rn5
    && rn5==rn6 && rn6 == rn7 && rn7 == rn8 && rn8 == rn9) {
    if (rn1 == 7) {
        budget += (bet*99);
        cout<<"All 777s earn 100x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*100)<<endl;
        numWin++;
        win+=(bet*99);

    }
    else {
        budget += (bet*49);
        cout<<"All same number earns 50x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*50)<<endl;
        numWin++;
        win+=(bet*49);
    }
}
else if (rn1==rn2 && rn2==rn3 && rn3==rn4 && rn4 == rn6
    && rn7==rn8 && rn8 == rn9) {
    if (rn1 == 7) {
        budget += (bet*14);
        cout<<"Square Pattern 777s earn 15x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*15)<<endl;
        numWin++;
        win+=(bet*14);
    }
    else {
        budget += (bet*9);
        cout<<"Square Pattern number earns 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
}
else if (rn2==rn4 && rn4==rn5 && rn5==rn6 && rn6==rn8) {
    if (rn2 == 7) {
        budget += (bet*14);
        cout<<"Cross Pattern 777s earn 15x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*15)<<endl;
        numWin++;
        win+=(bet*14);
    }
    else {
        budget += (bet*9);
        cout<<"Cross Pattern number earns 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
    }
}

```



```

        win+=(bet*9);
    }
}
else if (rn1==rn5 && rn5==rn9 && rn9==rn3 && rn3 == rn7) {
    if (rn1 == 7) {
        budget += (bet*9);
        cout<<"X Pattern 777s earn 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
    else {
        budget += (bet*4);
        cout<<"X Pattern number earns 5x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*5)<<endl;
        numWin++;
        win+=(bet*4);
    }
}
else if (rn4==rn5 && rn5==rn6) {
    if (rn4 == 7) {
        budget += (bet*9);
        cout<<"Straight 777s in the middle earn 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
    else {
        budget += bet;
        cout<<"Straight number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
else if (rn1==rn2 && rn2==rn3) {
    if (rn1 == 7) {
        budget += (bet*2);
        cout<<"Straight 777s earn 3x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*3)<<endl;
        numWin++;
        win+=(bet*2);
    }
    else {
        budget += bet;
        cout<<"Straight number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
}

```

```

    }
    else if (rn7==rn8 && rn8==rn9) {
        if (rn7 == 7) {
            budget += (bet*2);
            cout<<"Straight 777s earn 3x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*3)<<endl;
            numWin++;
            win+=(bet*2);
        }
        else {
            budget += bet;
            cout<<"Straight number earns 2x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*2)<<endl;
            numWin++;
            win+=bet;
        }
    }
    else if (rn1==rn5 && rn5==rn9) {
        if (rn1 == 7) {
            budget += (bet*2);
            cout<<"Diagonal 777s earn 3x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*3)<<endl;
            numWin++;
            win+=(bet*2);
        }
        else {
            budget += bet;
            cout<<"Diagonal number earns 2x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*2)<<endl;
            numWin++;
            win+=bet;
        }
    }
    else if (rn7==rn5 && rn5==rn3) {
        if (rn7 == 7) {
            budget += (bet*2);
            cout<<"Diagonal 777s earn 3x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*3)<<endl;
            numWin++;
            win+=(bet*2);
        }
        else {
            budget += bet;
            cout<<"Diagonal number earns 2x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*2)<<endl;
            numWin++;
            win+=bet;
        }
    }
    else {
        budget -= bet;
        cout<<"Sorry, you lost $"<<bet<<endl;
    }
}

```

```

        numLose++;
        loss+=bet;
    }

    if (play==g8mBon) {
        float added =
bonCash(added,bonCsh1,bonCsh2,bonCsh3,bonCsh4,loss,g8mBon,play,minVal);
        budget+=added;
        cout<<"\nCongratulations!\n"
        <<"You earned game bonus for reaching "<<g8mBon<<" number of
spins!"<<endl;

        cout<<"$"<<added<<" is added to your cash amount!"<<endl;
        g8mBon+=g8mBon;
        bonWin+=added;
    }

    budget = rndOffB(budget, rndOff);
    status(budget, name);
}
else {
    cout<<"\nYou can only enter numbers!\n"<<endl;
}
} while (budget>minVal && bet<minVal && budget!=minVal);
}

else if (choice == 'C' || choice == 'c') {
    if (play == minVal) {
        budget -= cshBon;
        bonWin -=cshBon;
        cout<<"\nNOTICE: Cash bonus is deducted because you have not played
yet!"<<endl;

        x = false;
    }
    else x=false;
}
else { cout<<"\nWarning: Enter only either 'S' or 'C' \n"<<endl; }
} while ((choice == 'S' || choice == 's') && x && budget != minVal);
}

else if (budget==minVal) {
    do {
        cout<<"You have used up all your funds. Would you like to insert again? "<<endl;
        cout<<"    Y - Yes    N - No "<<endl<<endl;
        cout<<"CHOICE: ";
        cin>>load;

        switch (load) {
            case 'y':
            case 'Y': {
                do {
                    cout<<"\nInsert Cash: $";

```

```

        cin>>add;

        while (cin.fail()) {
            cout<<"\nYou must only enter a number!"<<endl<<endl;
            cout<<"Insert Cash: $";
            cin.clear();
            getline(cin, dummy);
            cin>>add;
        }

        if (add > minVal) budget += add;
        else cout<<"Value must be greater than 0!"<<endl;

        cshIns+=add;

        budget = rndOffB(budget, rndOff);
        status(budget, name);

        } while (add <= minVal);
        z = false;
    } break;

    case 'n':
    case 'N': z = false;
        x = false;
        break;

    default: cout<<"\nYou can only enter 'Y' or 'N'\n"<<endl;
        break;
    }
} while (z);
}

else x=false;

} while (x);
y = false;
} break;

case 'm':
case 'M': {
    cout<<"\nWELCOME TO THE NUMBER GUESS MINI-GAME!\n"<<endl;
    cout<<"Rules: You must input a number and if it exists in the array for\n"
        <<" 3 times, you win $5 to be added to your current money. You will\n"
        <<" be able to cash it out if you proceed to play the slot machine\n"
        <<" after the mini-game. If you don't, the money will remain unless\n"
        <<" you choose not to play the game."<<endl<<endl;
    do {
        cout<<"PLAY?  P - PLAY    C - CANCEL"<<endl<<endl;
        cout<<"CHOICE: ";
        cin>>game;
    }
}

```

```

filArray(array, MINSIZE);

int findN;

if (game == 'p' || game == 'P') {
    cout<<"\nInput a number from 10-99!\n\n";
    cout<<"Number: ";
    cin>>findN;

    while (cin.fail()) {
        cout<<"\nYou must only enter a number!"<<endl<<endl;
        cout<<"Number: ";
        cin.clear();
        getline(cin, dummy);
        cin>>findN;
    }

    mrkSrt(array, MINSIZE);
    find(array, MINSIZE, found, findN);
    cout<<endl;
    prntAry(array, MINSIZE);
    cout<<"\nNumber "<<findN<<" occured "<<found<<" times.\n"<<endl

    if (found>=3) {
        cout<<"Congratulations! $5 is added to the current money!\n"<<endl;
        budget+=5;
        minCsh+=5;
        bonWin+=minCsh;
    }
    else {
        cout<<"Sorry, you didn't win. Try again!\n"<<endl;
        budget=budget;
    }

    found = 0;
}

else if (game == 'C' || game == 'c') {
    cout<<"\nExiting mini game..."<<endl<<endl;
}
else cout<<"\n\nWarning: You can only enter 'P' or 'C'!\n"<<endl;

} while (game == 'p' || game == 'P');

y=false; //Exit mini-game back to main menu
} break;

case 'n':
case 'N': {
    cout<<"\nGoodbye!"<<endl;

```

```

        budget = 0;
        y = false;
    } break;

case 's':
case 'S': {
    cout<<endl;
    cout<<"Staff Login"<<endl<<endl;
    cout<<"Employee Name: ";
    cin>>empName;
    cout<<"Input Password: ";
    cin>>input;
    cout<<"\nAccessing database..."<<endl<<endl;

    if (valPass(input, empName)) {
        cout<<"\n\tPlayer List \t\t\t\t\tTotal Spins \t\tTotal Cash \t\t\tCash Bonus \t\t\t\t\tCash Won \t\t\t\t\tCash Loss"<<endl;
        cout<<"\t-----"
        -----"<<endl;
        for (int cHere = 0; cHere < ROWS; cHere++) {
            cout<<setw(19);
            cout<<names[cHere]<<setw(19)<<spins[cHere]<<setw(21);
            for (int iHere = 0; iHere < COLS; iHere++) {
                cout<<data[cHere][iHere]<<setw(20);
            }
            cout<<endl;
        }
        cout<<"\nRESET? Y - YES N - NO"<<endl<<endl;
        cout<<"CHOICE: ";
        cin>>reset;
        if (reset == 'Y' || reset == 'y') {
            cout<<"\nResetting everything!"<<endl;
            y=false;
            w=false;
            x=true;
        }
        else if (reset == 'N' || reset == 'n') {
            cout<<"\nExiting staff settings..."<<endl<<endl;
            y=false;
            x=true;
        }
        else cout<<"\nWarning: You can only enter 'Y' or 'N'!\nRepeat process to access!"<<endl;
    }
    else {
        cout<<"Log in failed!\n\n";
        x=true; y=false; }
    } break;

default: {
    cout<<"\nWarning: You can only enter 'Y' or 'N' or 'M'!\n"<<endl; y=false;
    } break;

```



```

    }
} while (y);

    cout<<fixed<<setprecision(2)<<showpoint;

if (x == false) {
    int num=0;
    totCash = cshIns + bonWin + win;
    spins[count] = play;
    data[count][num] = totCash; num++;
    data[count][num] = bonWin; num++;
    data[count][num] = win; num++;
    data[count][num] = loss; num++;
    count++;
    num-=4;
    plCount++;

    for (int p = 1; p <= 1; p++) {
        if (budget>minVal) {
            voucher(name, budget);
            do {
                askRep();
                cin>>report;

                if (report == 'Y' || report == 'y') {
                    gameRep(name, report, play, numWin, numLose, cshIns, win, bonWin, totCash,
                        loss, winPer, lossPer);
                    rep = false;
                    x = true;
                    resetA(play, numWin, numLose, cshIns, win, bonWin, totCash, loss, winPer, lossPer,
budget);
                }
                else if (report == 'N' || report == 'n') {
                    gameRep(name, report, play, numWin, numLose, cshIns, win, bonWin, totCash,
                        loss, winPer, lossPer);
                    rep = false;
                    x=true;
                    resetA(play, numWin, numLose, cshIns, win, bonWin, totCash, loss, winPer, lossPer,
budget);
                }
                else cout<<"\nWarning: You can only enter 'Y' or 'N'!"<<endl;
            } while (rep);
        }

        else if (budget <=0) {
            cout<<"\nSorry, you didn't win any amount this time."<<endl;

            do {
                askRep();
                cin>>report;

```

```

        if (report == 'Y' || report == 'y') {
            gameRep(name, report, play, numWin, numLose, cshIns, win, bonWin, totCash,
                    loss, winPer, lossPer);
            rep = false;
            x=true;
            resetA(play, numWin, numLose, cshIns, win, bonWin, totCash, loss, winPer, lossPer,
budget);
        }

        else if (report == 'N' || report == 'n') {
            gameRep(name, report, play, numWin, numLose, cshIns, win, bonWin, totCash,
                    loss, winPer, lossPer);
            rep = false;
            x = true;
            resetA(play, numWin, numLose, cshIns, win, bonWin, totCash, loss, winPer, lossPer,
budget);
        }
        else cout<<"\nWarning: You can only enter 'Y' or 'N'!"<<endl;
    } while (rep);
}
}
}

else if (begin == 'n' || begin == 'N' || begin == 'y' || begin == 'Y'&&play<=0) {
    cout<<"\nPlay the game next time!\n"<<endl;
}
} while (w);
return 0;
}

```

---

```

if (budget >= 0 && budget <= 20000) {
    int exCshBn = cshBon*rndOff+0.5;
    cshBon = exCshBn/100.0f;

    budget += cshBon;
    cout<<"\nThank you for choosing to play the game!"<<endl;
    cout<<"We added $"<<cshBon<<" to your capital as cash bonus."<<endl<<endl;
    cout<<"Good luck ^_^"<<endl;
    bonWin+=cshBon;
}
int exBud = budget*rndOff+0.5;
budget = exBud/100.0f;

cout<<endl
    <<"-----"<<endl;
cout<<"Player: "<<name<<endl;
cout<<"Current Money: $"<<budget<<endl;
cout<<"-----"<<endl<<endl;

```

[illegible]

```

        (g8mBon<=300)?(loss*bonCsh3):
        (g8mBon>=301)?(loss*bonCsh4):minVal;
    }

    rn1 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn2 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn3 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn4 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn5 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn6 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn7 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn8 = (rand()%(8-4+1))+4; //Value range from 4 to 8
    rn9 = (rand()%(8-4+1))+4; //Value range from 4 to 8

    cout<<endl;
    cout<<"\t"<<rn1<<" "<<rn2<<" "<<rn3<<endl;
    cout<<"\t"<<rn4<<" "<<rn5<<" "<<rn6<<endl;
    cout<<"\t"<<rn7<<" "<<rn8<<" "<<rn9<<endl<<endl;

    if (rn1==rn2 && rn2==rn3 && rn3==rn4 && rn4 == rn5
        && rn5==rn6 && rn6 == rn7 && rn7 == rn8 && rn8 == rn9) {
        if (rn1 == 7) {
            budget += (bet*99);
            cout<<"All 777s earn 100x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*100)<<endl;
            numWin++;
            win+=(bet*99);
        }
        else {
            budget += (bet*49);
            cout<<"All same number earns 50x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*50)<<endl;
            numWin++;
            win+=(bet*49);
        }
    }
    else if (rn1==rn2 && rn2==rn3 && rn3==rn4 && rn4 == rn6
        && rn7==rn8 && rn8 == rn9) {
        if (rn1 == 7) {
            budget += (bet*14);
            cout<<"Square Pattern 777s earn 15x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*15)<<endl;
            numWin++;
            win+=(bet*14);
        }
        else {
            budget += (bet*9);
            cout<<"Square Pattern number earns 10x bet!"<<endl;
            cout<<"Congratulations for winning $"<<(bet*10)<<endl;
            numWin++;
            win+=(bet*9);
        }
    }
}

```

```

    }
}
else if (rn2==rn4 && rn4==rn5 && rn5==rn6 && rn6==rn8) {
    if (rn2 == 7) {
        budget += (bet*9);
        cout<<"Cross Pattern 777s earn 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
    else {
        budget += (bet*4);
        cout<<"Cross Pattern number earns 5x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*5)<<endl;
        numWin++;
        win+=(bet*4);
    }
}
else if (rn1==rn5 && rn5==rn9 && rn9==rn3 && rn3 == rn7) {
    if (rn1 == 7) {
        budget += (bet*9);
        cout<<"X Pattern 777s earn 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
    else {
        budget += (bet*4);
        cout<<"X Pattern number earns 5x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*5)<<endl;
        numWin++;
        win+=(bet*4);
    }
}
else if (rn4==rn5 && rn5==rn6) {
    if (rn4 == 7) {
        budget += (bet*9);
        cout<<"Straight 777s in the middle earn 10x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*10)<<endl;
        numWin++;
        win+=(bet*9);
    }
    else {
        budget += bet;
        cout<<"Straight number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
else if (rn1==rn2 && rn2==rn3) {

```

```

if (rn1 == 7) {
    budget += (bet*2);
    cout<<"Straight 777s earn 3x bet!"<<endl;
    cout<<"Congratulations for winning $"<<(bet*3)<<endl;
    numWin++;
    win+=(bet*2);
}
else {
    budget += bet;
    cout<<"Straight number earns 2x bet!"<<endl;
    cout<<"Congratulations for winning $"<<(bet*2)<<endl;
    numWin++;
    win+=bet;
}
}
else if (rn7==rn8 && rn8==rn9) {
    if (rn7 == 7) {
        budget += (bet*2);
        cout<<"Straight 777s earn 3x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*3)<<endl;
        numWin++;
        win+=(bet*2);
    }
    else {
        budget += bet;
        cout<<"Straight number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
else if (rn1==rn5 && rn5==rn9) {
    if (rn1 == 7) {
        budget += (bet*2);
        cout<<"Diagonal 777s earn 3x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*3)<<endl;
        numWin++;
        win+=(bet*2);
    }
    else {
        budget += bet;
        cout<<"Diagonal number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
else if (rn7==rn5 && rn5==rn3) {
    if (rn7 == 7) {
        budget += (bet*2);
        cout<<"Diagonal 777s earn 3x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*3)<<endl;
    }
}

```



```

        numWin++;
        win+=(bet*2);
    }
    else {
        budget += bet;
        cout<<"Diagonal number earns 2x bet!"<<endl;
        cout<<"Congratulations for winning $"<<(bet*2)<<endl;
        numWin++;
        win+=bet;
    }
}
else {
    budget -= bet;
    cout<<"Sorry, you lost $"<<bet<<endl;
    numLose++;
    loss+=bet;
}

exBud = budget*rndOff+0.5;
budget = exBud/100.0f;

cout<<"\n"
    <<"-----"<<endl;
cout<<"Player: "<<name<<endl;
cout<<"Current Money: $"<<budget<<endl;
cout<<"-----"<<endl<<endl;
}
else {
    cout<<"\nYou can only enter numbers!\n"<<endl;
}
} while (budget>minVal && bet<minVal && budget!=minVal);
}

else if (choice == 'C' || choice == 'c') {
    if (play == minVal) {
        budget -= cshBon;
        bonWin -=cshBon;
        cout<<"\nNOTICE: Cash bonus is deducted because you have not played
yet!"<<endl;
        x = false;
    }
    else x=false;
}
else { cout<<"\nWarning: Enter only either 'S' or 'C' \n"<<endl; }
} while ((choice == 'S' || choice == 's') && x && budget != minVal);
}

else if (budget==minVal) {
    do {
        cout<<"You have used up all your funds. Would you like to insert again? "<<endl;
        cout<<"    Y - Yes    N - No  "<<endl<<endl;
        cout<<"CHOICE: ";
    }

```

```

cin>>load;

switch (load) {
    case 'y':
    case 'Y': {
        do {
            cout<<"\nInsert Cash: $";
            cin>>add;

            while (cin.fail()) {
                cout<<"\nYou must only enter a number!"<<endl<<endl;
                cout<<"Insert Cash: $";
                cin.clear();
                getline(cin, dummy);
                cin>>add;
            }

            if (add > minVal) budget += add;
            else cout<<"Value must be greater than 0!"<<endl;

            cshIns+=add;
            exBud = budget*rndOff+0.5;

            budget = exBud/100.0f;

            cout<<"\n"
                <<"-----"<<endl;
            cout<<"Player: "<<name<<endl;
            cout<<"Current Money: $"<<budget<<endl;
            cout<<"-----"<<endl<<endl;

        } while (add <= minVal);
        z = false;
    } break;

    case 'n':
    case 'N': z = false;
            x = false;
            break;

    default: cout<<"\nYou can only enter 'Y' or 'N'\n"<<endl;
            break;
        }
    } while (z);
}

else x=false;

} while (x);
y = false;
} break;

```

```

    case 'n':
    case 'N': {
        cout<<"\nGoodbye!"<<endl;
        y = false;
    } break;

    default: {
        cout<<"\nWarning: You can only enter 'Y' or 'N'!\n"<<endl;
    } break;
}
} while (y);

cout<<fixed<<setprecision(2)<<showpoint;

if (x == false) {

    totCash = cshIns + bonWin + win;

    for (int p = 1; p <= 1; p++) {
        if (budget>minVal) {
            cout<<"\nCashout voucher printing..."<<endl<<endl;

            unsigned short ticket = (rand()%(9999-1000+1))+1000;
            unsigned short v1 = (rand()%(99-10+1))+10;
            unsigned short v2 = (rand()%(9999-1000+1))+1000;
            unsigned short v3 = (rand()%(9999-1000+1))+1000;
            unsigned short v4 = (rand()%(9999-1000+1))+1000;
            unsigned short v5 = (rand()%(9999-1000+1))+1000;
            unsigned int mNum = (rand()%(4294967294-1000000000+1))+1000000000;

            cout<<"\t-----"<<endl;
            cout<<"\t CSC5 CASINO RIVERSIDE, CA"<<setw(27)<<ticket<<endl<<endl;
            cout<<"\t\t\t CASHOUT VOUCHER"<<endl<<endl;
            cout<<"\t\t\tFOR "<<name<<endl<<endl;
            cout<<"\t\t Validation"<<setw(5)<<v1<<"-"<<v2<<"-"<<v3<<"-"<<v4<<"-"<<v5<<endl;
            cout<<"\t\t " <<dt<<endl<<fixed<<setprecision(2)<<showpoint;
            cout<<"\t\t " <<setw(5)<<"$"<<budget<<endl<<endl;
            cout<<"\t Ticket void after 30 days"<<setw(17)<<"Machine #"<<mNum<<endl;
            cout<<"\t-----"<<endl<<endl;
            cout<<"\tNOTE: Bring your ID with you to cash it out to verify \n"
                <<"\t name in the voucher."<<endl;

            do {
                cout<<"\nWould you like to see your game report?"<<endl
                    <<"      Y - YES      N - NO"<<endl<<endl;
                cout<<"CHOICE: ";
                cin>>report;

                if (report == 'Y' || report == 'y') {
                    //
                    cout<<"\n\tSLOT MACHINE GAME REPORT"<<endl<<endl;

```

```

    cout<<"Number of games played: "<<play<<endl;
    cout<<"Number of wins: "<<numWin<<endl;
    cout<<"Number of losses: "<<numLose<<endl<<endl;
    cout<<"Total amount of cash inserted: $"<<cshIns<<endl;
    cout<<"Total amount of cash won: $"<<win<<endl;
    cout<<"Total amount of cash bonus: $"<<bonWin<<endl<<endl;
    cout<<"Total amount of cash involved: $"<<totCash<<endl;
    cout<<"Total amount of cash loss: $"<<loss<<endl<<endl;

    winPer = (static_cast<float>(numWin)/play)*PERCENT;
    lossPer = (static_cast<float>(numLose)/play)*PERCENT;

    cout<<"Win Percentage: "<<winPer<<"%"<<endl;
    cout<<"Loss Percentage: "<<lossPer<<"%"<<endl;

    cout<<"\nThank you for playing the game! See you again next time!"<<endl<<endl;

    rep = false;
}
else if (report == 'N' || report == 'n') {
    ofstream out;
    out.open(name+".dat");

    out<<"\n\tSLOT MACHINE GAME REPORT"<<endl<<endl;
    out<<"Number of games played: "<<play<<endl;
    out<<"Number of wins: "<<numWin<<endl;
    out<<"Number of losses: "<<numLose<<endl<<endl;
    out<<"Total amount of cash inserted: $"<<cshIns<<endl;
    out<<"Total amount of cash won: $"<<win<<endl;
    out<<"Total amount of cash bonus: $"<<bonWin<<endl<<endl;
    out<<"Total amount of cash involved: $"<<totCash<<endl;
    out<<"Total amount of cash loss: $"<<loss<<endl<<endl;

    winPer = (static_cast<float>(numWin)/play)*PERCENT;
    lossPer = (static_cast<float>(numLose)/play)*PERCENT;

    out<<"Win Percentage: "<<winPer<<"%"<<endl;
    out<<"Loss Percentage: "<<lossPer<<"%"<<endl;
    cout<<"\nGame report is saved to a file."<<endl;
    cout<<"\nThank you for playing! See you again!\n"<<endl;

    out.close();
    rep = false;
}

    else cout<<"\nWarning: You can only enter 'Y' or 'N'!"<<endl;
} while (rep);
}

else if (budget <=0) {
    cout<<"\nSorry, you didn't win any amount this time."<<endl;

```

```

do {
cout<<"\nWould you like to see your game report?"<<endl
    <<"      Y - YES    N - NO"<<endl<<endl;
cout<<"CHOICE: ";
cin>>report;

if (report == 'Y' || report == 'y') {
    cout<<"\n\tSLOT MACHINE GAME REPORT"<<endl<<endl;
    cout<<"Number of games played: "<<play<<endl;
    cout<<"Number of wins: "<<numWin<<endl;
    cout<<"Number of losses: "<<numLose<<endl<<endl;
    cout<<"Total amount of cash inserted: $"<<cshIns<<endl;
    cout<<"Total amount of cash won: $"<<win<<endl;
    cout<<"Total amount of cash bonus: $"<<bonWin<<endl<<endl;
    cout<<"Total amount of cash involved: $"<<totCash<<endl;
    cout<<"Total amount of cash loss: $"<<loss<<endl<<endl;

    winPer = (static_cast<float>(numWin)/play)*PERCENT;
    lossPer = (static_cast<float>(numLose)/play)*PERCENT;

    cout<<"Win Percentage: "<<winPer<<"%"<<endl;
    cout<<"Loss Percentage: "<<lossPer<<"%"<<endl;

    cout<<"\nThank you for playing the game! See you again next time!"<<endl<<endl;

    rep = false;
}
else if (report == 'N' || report == 'n') {
    ofstream out;
    out.open(name+".dat");

    out<<"\n\tSLOT MACHINE GAME REPORT"<<endl<<endl;
    out<<"Number of games played: "<<play<<endl;
    out<<"Number of wins: "<<numWin<<endl;
    out<<"Number of losses: "<<numLose<<endl<<endl;
    out<<"Total amount of cash inserted: $"<<cshIns<<endl;
    out<<"Total amount of cash won: $"<<win<<endl;
    out<<"Total amount of cash bonus: $"<<bonWin<<endl<<endl;
    out<<"Total amount of cash involved: $"<<totCash<<endl;
    out<<"Total amount of cash loss: $"<<loss<<endl<<endl;

    winPer = (static_cast<float>(numWin)/play)*PERCENT;
    lossPer = (static_cast<float>(numLose)/play)*PERCENT;

    out<<"Win Percentage: "<<winPer<<"%"<<endl;
    out<<"Loss Percentage: "<<lossPer<<"%"<<endl;
    cout<<"\nGame report is saved to a file."<<endl;
    cout<<"\nThank you for playing! See you again!\n"<<endl;

    out.close();
}
}

```

```
        rep = false;
    }
    else cout<<"\nWarning: You can only enter 'Y' or 'N'!"<<endl;
} while (rep);
}
}
}
else cout<<"\nPlay the game next time!"<<endl

return 0;
}
```

