Project 2

<Hide And Seek-Expanded>

CSC-5 46687
Name: Triet Huynh
Date: 07/30/2022

TABLE OF CONTENTS

Introduction

Title: Hide and Seek
The project is a simple hide-and-seek game that is available in single-player and multiplayer modes.
Upon starting, the player(s) is introduced to a board of nine slots and will have to guess which one is the correct slot that the machine is hiding.

Summary

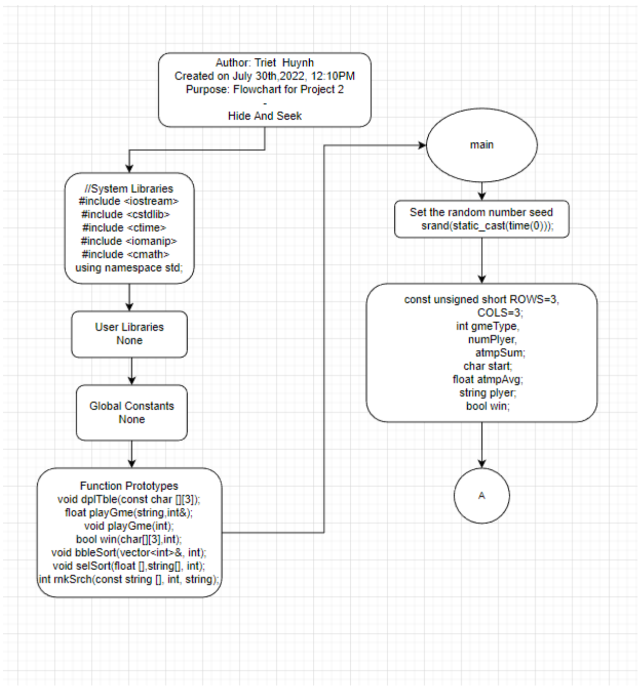
Project Size: Around 380 lines.
The number of variables: about 15
The project took about two days to code and a few days to produce the idea. I did not have much problem with coding after having steps laid out and pseudocode written.
How the game proceeds:
-Originally with single-player mode, the player will have three rounds and as many attempts as it takes to find the hiding spot each round. After each attempt, the game will display the game board with the slot updated as “0” if guessed correctly and “X” otherwise. The program will store the attempts each round in a file named after the player as a record, and it will calculate and announce the average attempts it took per round to the player.
-The project has been expanded into project 2 as we introduce a multiplayer mode and let the players compete to see who wins with fewer attempts on average to locate the hiding slot. A ranking system also has been added to show the players’ ranks based on their average attempts.

CROSS REFERENCE FOR PROJECT 2

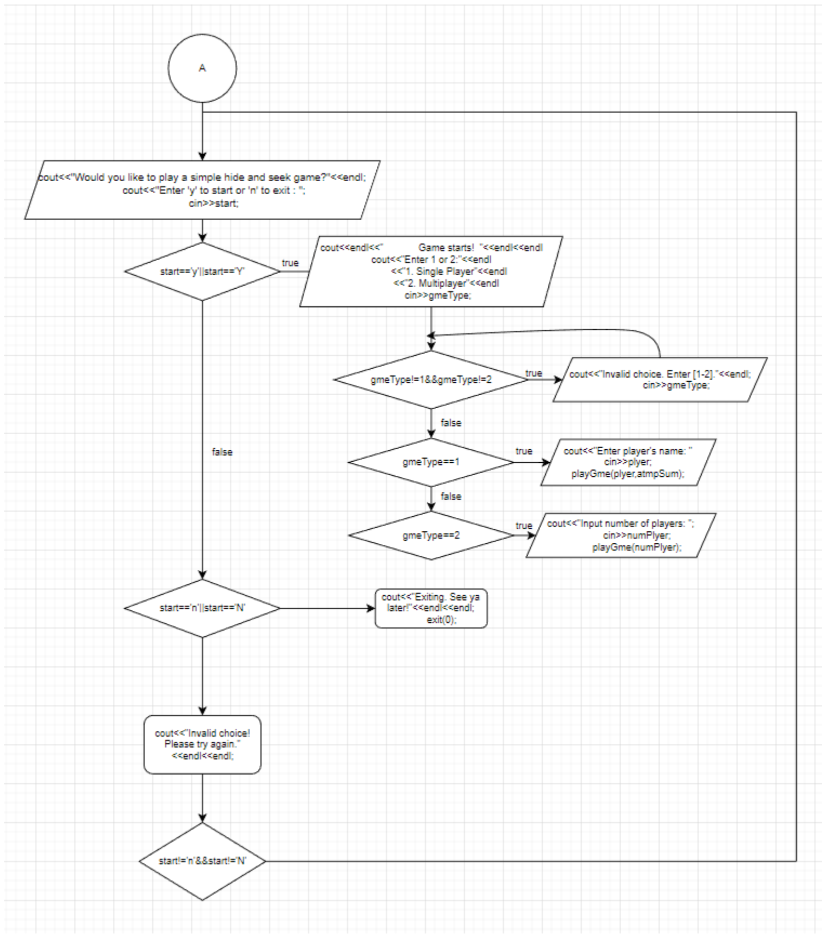
Chapter	Topic	Where Line
2	cout	51,52,55,56,etc
	libraries	9,10,11,12,13,14,15
	variables/literals	38-47,etc
	Identifiers	38-47,etc
	Integers	40,41,42,etc
	Characters	43,112,285,286
	Strings	45,278,279
	Floats No Doubles	44,104,281,348,etc
	Bools	46,267
	Variables 7 characters or less	all
	Comments 20%+	throughout
	Named Constants	97,98
3	cin	58,66,70,etc
	Math Expression	112,261
	Type Casting	261
	Formatting output	262,303,304,305
	Strings	45,278,279
	Math Library	261
4	Relational Operators	53,60,64,68,110,etc
	if	270,338,354,etc
	If-else	121,123,127,133,etc
	Nesting	53,64,68,etc
	If-else-if	53,73,76,etc
	Logical operators	53,60,64,68,110,etc
	Validating user input	59,120,310,325
	Switch	125
5	Increment/Decrement	124,268,269,289,302,etc
	While	60,257,311,236,374
	Do-while	49,117,315
	For loop	110,268,269,etc
	Files input/output both	107,252,254,255,256,000
6	Function Prototypes	24-30
	Pass by Value	67,71,293
	return	264,271,274,382
	returning boolean	271,274
	Global Variables	NONE
	default arguments	277
	pass by reference	25,28,67,293,298,334
	overloading	25,26,96,277
	exit() function	75
7	Single Dimensioned Arrays	282,283
	Parralel Arrays	282,283
	Single Dimensioned as Function Arguments	299,318
	2 Dimensioned Arrays	111
	STL Vectors	284
	Passing Assays to and from Functions	267,299,318,347,etc
	Passing Vectors to and from Functions	298,347
8	Bubble Sort	298,334
	Selection Sort	299,347
	Linear or Binary Search	318,370

Pseudocodes & Flowcharts

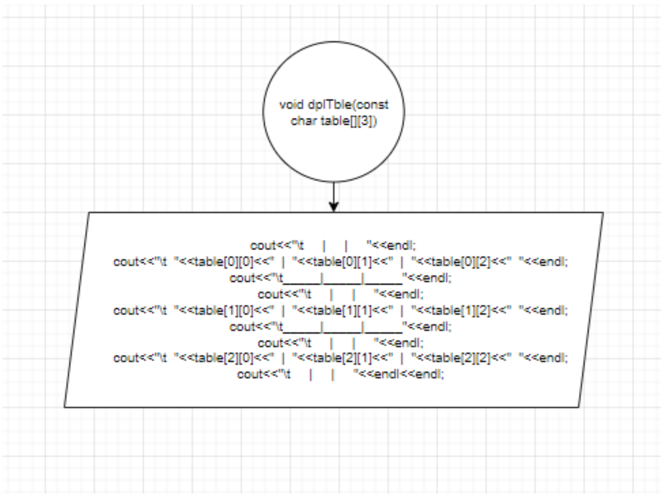
```
-Opening Comments (name, author, date, purpose)

-Libraries, namespace std

-Function Prototypes

-Main Function
 Set random number seed
 Declare Variables needed for main
```



```
Display games start message and
            Get user input into 'start'

 If start is 'y' or 'Y'
   Prompt user for single or multiplayer
      Verify input with while loop
      If single player,
            Prompt user for their name &
            call single-player game function
      If multi-player,
            Prompt user for number of players
            Call multi-player game function

Else if start is 'n' or 'N'
  Display goodbye message and exit

Else
  Invalid input

Repeat this in a do-while loop until
            Player inputs 'n' or 'N'
```

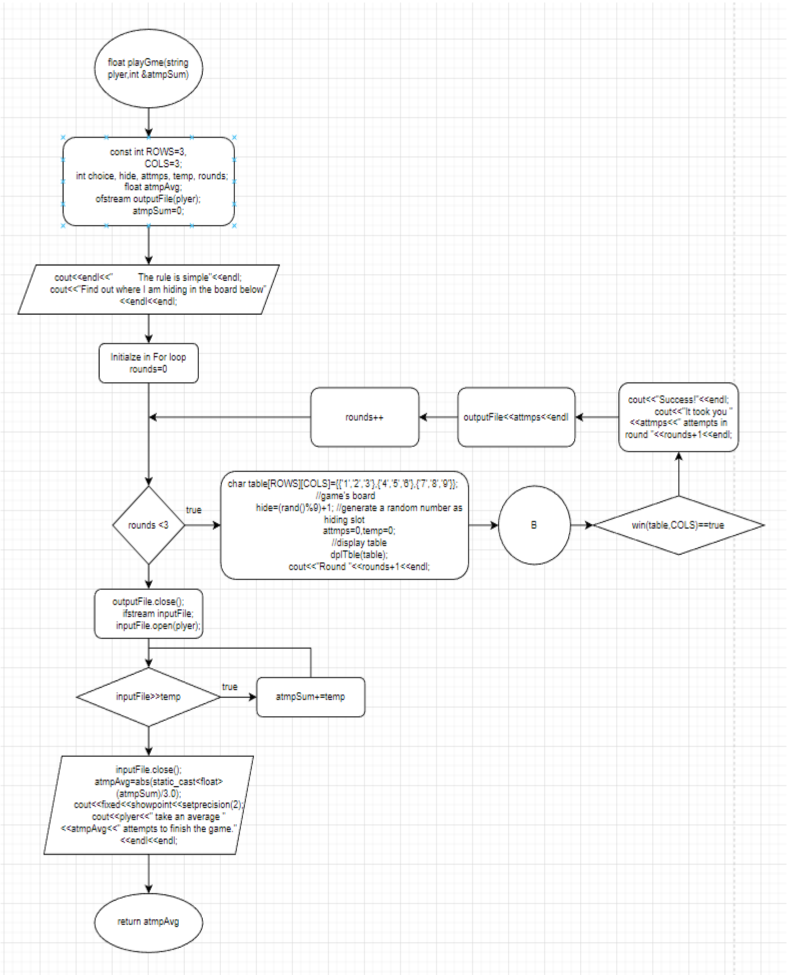

A
cout<<"Would you like to play a simple hide and seek game?"<<endl;
cout<<"Enter 'y' to start or 'n' to exit : ";
cin>>start;
start=='y'||start=='Y'
true
cout<<endl<<"   Game starts! "<<endl<<endl
cout<<"Enter 1 or 2:"<<endl
<<"1. Single Player"<<endl
<<"2. Multiplayer"<<endl
cin>>gmeType;
gmeType!=1&&gmeType!=2
true
cout<<"Invalid choice. Enter [1-2]."<<endl;
cin>>gmeType;
false
gmeType==1
true
cout<<"Enter player's name: "
cin>>plyer;
playGme(plyer,atmpSum);
false
gmeType==2
true
cout<<"Input number of players: ";
cin>>numPlyer;
playGme(numPlyer);
false
start=='n'||start=='N'
cout<<"Exiting. See ya later!"<<endl<<endl;
exit(0);
cout<<"Invalid choice! Please try again."<<endl<<endl;
start!='n'&&start!='N'

```
-Display table function
    Pass in 2D table array and
          Output the elements in rows and cols
```



```
cout<<"\t    |    |    "<<endl;
cout<<"\t  "<<table[0][0]<<"  |  "<<table[0][1]<<"  |  "<<table[0][2]<<"  "<<endl;
cout<<"\t_____|_____|_____"<<endl;
cout<<"\t    |    |    "<<endl;
cout<<"\t  "<<table[1][0]<<"  |  "<<table[1][1]<<"  |  "<<table[1][2]<<"  "<<endl;
cout<<"\t_____|_____|_____"<<endl;
cout<<"\t    |    |    "<<endl;
cout<<"\t  "<<table[2][0]<<"  |  "<<table[2][1]<<"  |  "<<table[2][2]<<"  "<<endl;
cout<<"\t    |    |    "<<endl<<endl;
```

```
-Single player game function

  Declare and initialize variables needed

  Output game rule

  Open file to write data to

  Use for loop to run 3 game rounds:
     Generate a random number
          1-9 as hiding slot
     Set attempts to 0

     Display game table with slots
          1-9

     Initialize a do-while loop

       Prompt user for their guess input

       Validate input

          If input<1 and input>9
```


float playGme(string plyer,int &atmpSum)
Initialze in For loop rounds=0
rounds <3
true
B
win(table,COLS)==true
inputFile>>temp
true
return atmpAvg

```
const int ROWS=3,
        COLS=3;
int choice, hide, attmps, temp, rounds;
float atmpAvg;
ofstream outputFile(plyer);
atmpSum=0;
```

```
cout<<endl<<"        The rule is simple"<<endl;
cout<<"Find out where I am hiding in the board below"
<<endl<<endl;
```

```
char table[ROWS][COLS]={{'1','2','3'},{'4','5','6'},{'7','8','9'}};
//game's board
hide=(rand()%9)+1; //generate a random number as
hiding slot
attmps=0,temp=0;
//display table
dplTble(table);
cout<<"Round "<<rounds+1<<endl;
```

```
cout<<"Success!"<<endl;
cout<<"It took you "
<<attmps<<" attempts in
round "<<rounds+1<<endl;
```

```
outputFile<<attmps<<endl
```

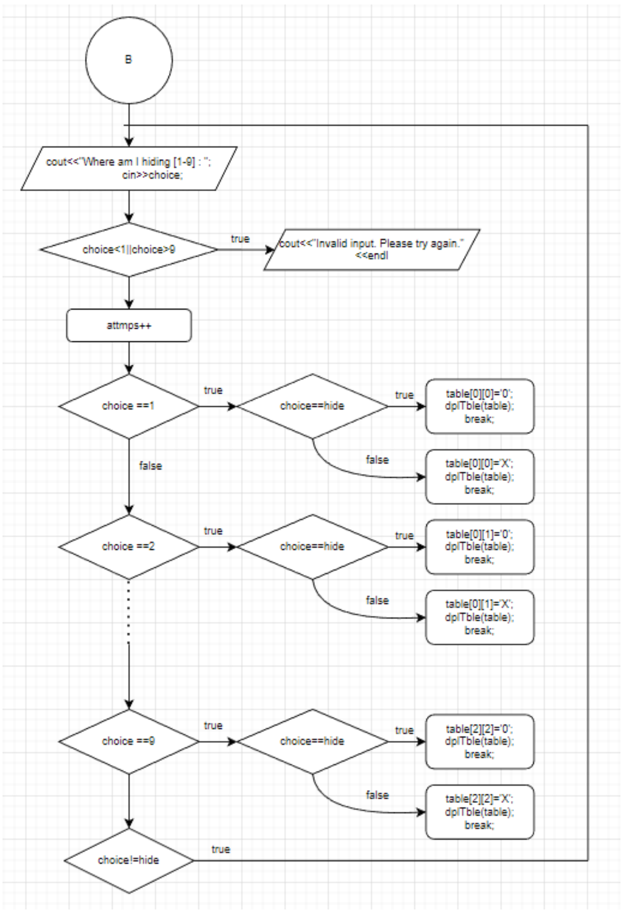
```
rounds++
```

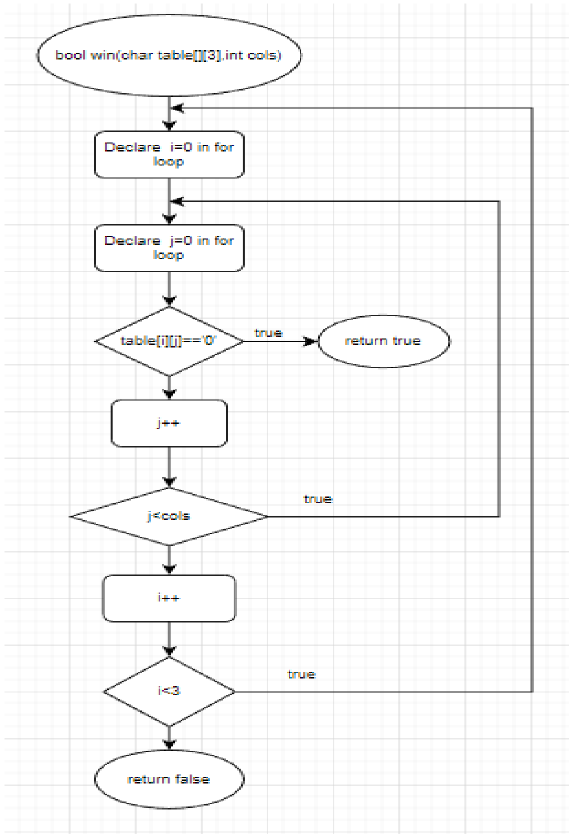
```
outputFile.close();
ifstream inputFile;
inputFile.open(plyer);
```

```
atmpSum+=temp
```

```
inputFile.close();
atmpAvg=abs(static_cast<float>
(atmpSum)/3.0);
cout<<fixed<<showpoint<<setprecision(2);
cout<<plyer<<" take an average "
<<atmpAvg<<" attempts to finish the game."
<<endl<<endl;
```

```
          Display error and prompt user to try again

        If input is in range 1-9,
          Increment attempt

          If input does not match hide,
                Replace slot with 'X'
                Display updated table

          If input matches hide,
                Replace slot with '0'
                Display updated table
                Display success message
                Round ends

      Loop these steps until user guesses
        Correctly and move onto next round

        Output number of attempts onto
                File after each round

Close output data file and open input data file

Input attempts of 3 rounds from file, then add
        And calculate the average attempts

Display average attempts

Return average attempts for float function type
```


B
cout<<"Where am I hiding [1-9] : "; cin>>choice;
choice<1||choice>9
true
cout<<"Invalid input. Please try again."<<endl
attmps++
choice ==1
true
choice==hide
true
table[0][0]='0'; dplTble(table); break;
false
table[0][0]='X'; dplTble(table); break;
false
choice ==2
true
choice==hide
true
table[0][1]='0'; dplTble(table); break;
false
table[0][1]='X'; dplTble(table); break;
choice ==9
true
choice==hide
true
table[2][2]='0'; dplTble(table); break;
false
table[2][2]='X'; dplTble(table); break;
choice!=hide
true



```
-Win function

Check each of array table's elements

 If any of the elements matches '0',
        return true

 Otherwise return false
```

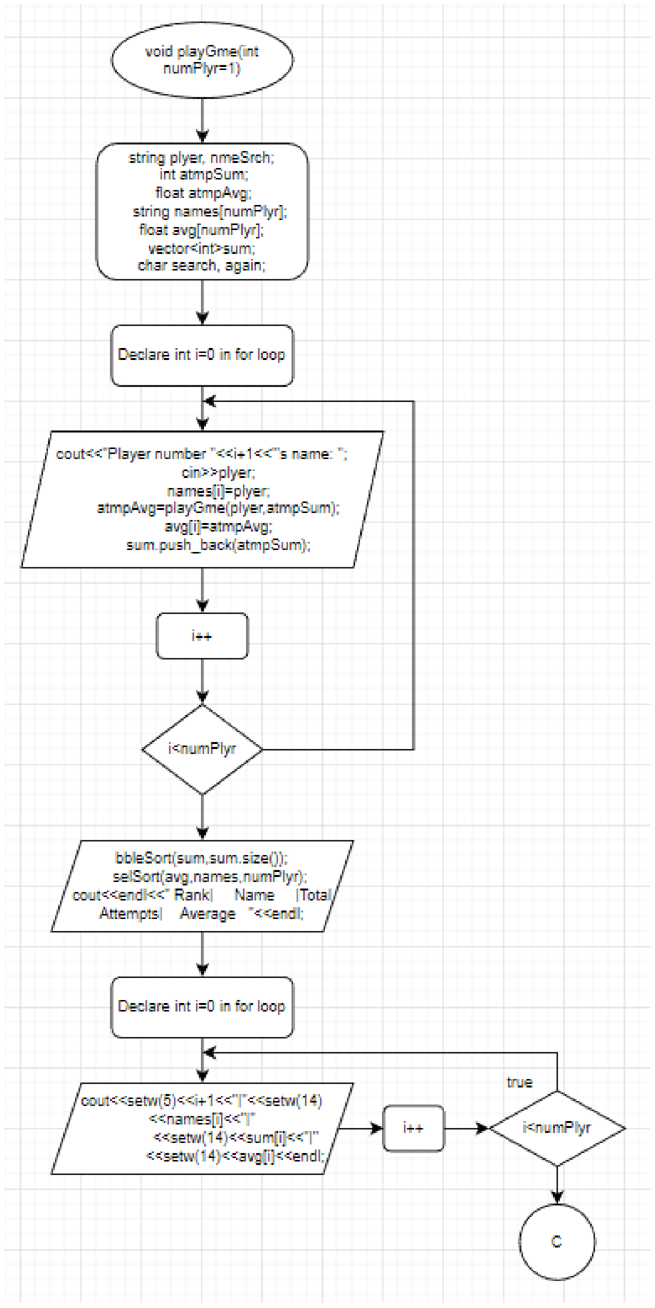
-Multiplayer function

Declare necessary variables, arrays and vector

Repeat the single-player game for each
of the players while inputting
their names, total scores and
average scores in arrays and vector

Sort the arrays using selection sort
and vector using bubble sort
in descending order
while making sure they are parallel

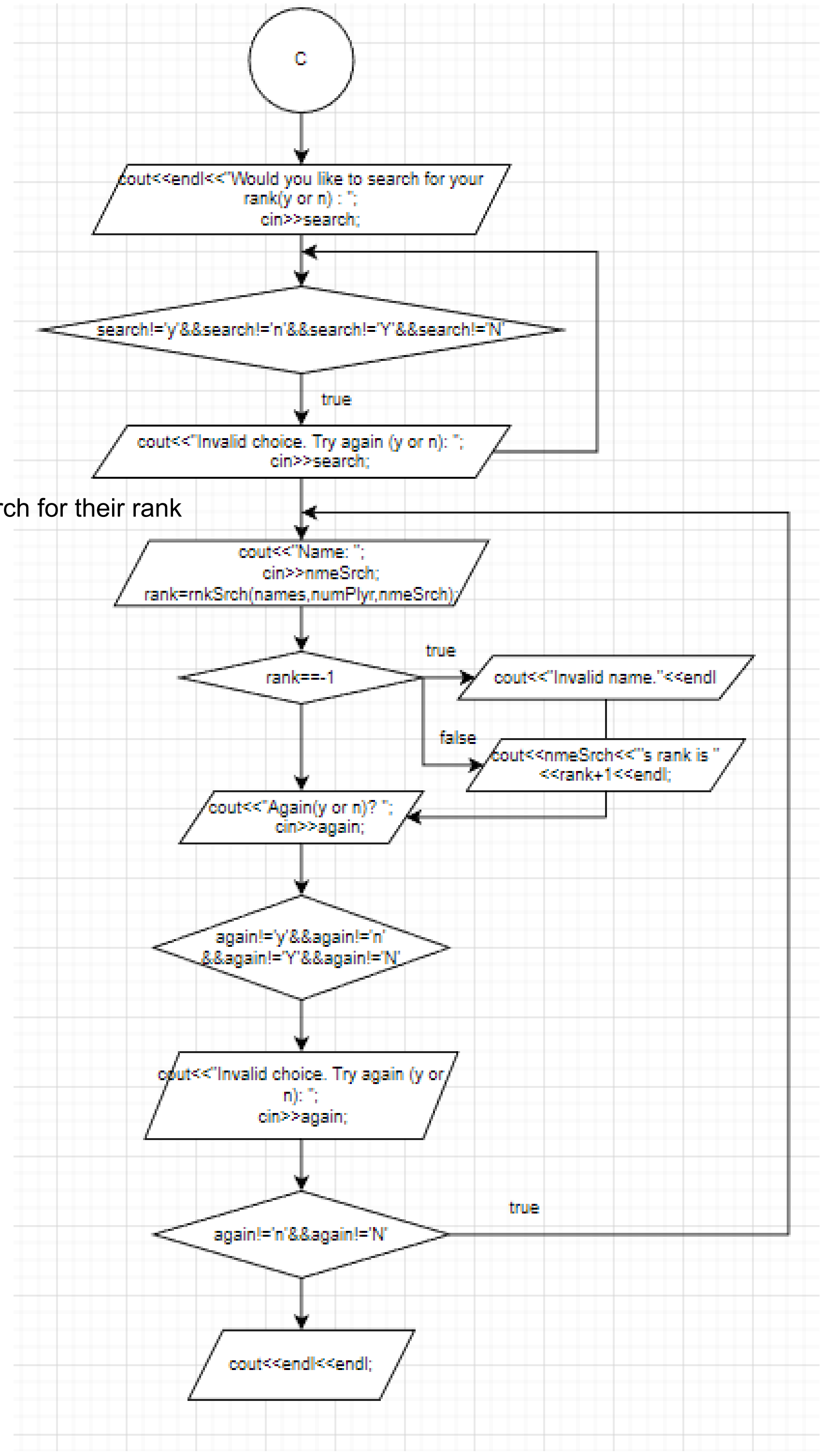
Display the ranking table


void playGme(int numPlyr=1)
string plyer, nmeSrch; int atmpSum; float atmpAvg; string names[numPlyr]; float avg[numPlyr]; vector<int>sum; char search, again;
Declare int i=0 in for loop
cout<<"Player number "<<i+1<<"'s name: "; cin>>plyer; names[i]=plyer; atmpAvg=playGme(plyer,atmpSum); avg[i]=atmpAvg; sum.push_back(atmpSum);
i++
i<numPlyr
bbleSort(sum,sum.size()); selSort(avg,names,numPlyr); cout<<endl<<" Rank| Name |Total Attempts| Average "<<endl;
Declare int i=0 in for loop
cout<<setw(5)<<i+1<<"|"<<setw(14) <<names[i]<<"|" <<setw(14)<<sum[i]<<"|" <<setw(14)<<avg[i]<<endl;
i++
true
i<numPlyr
C

Prompt users whether any wants to search for their rank
validate user input

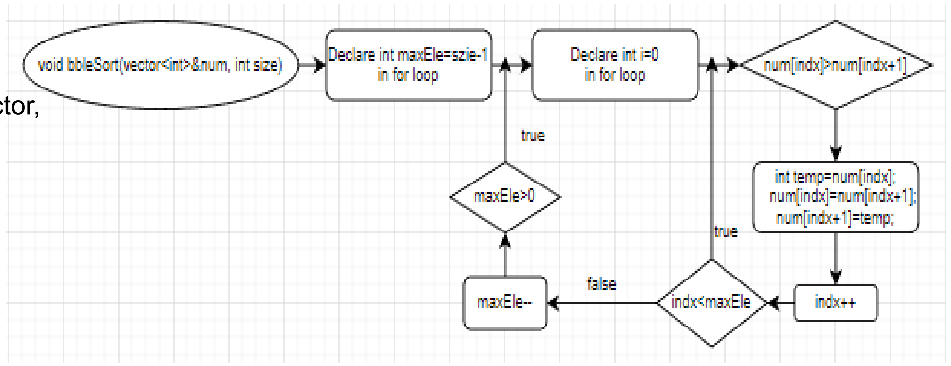
Prompt user for their name and
validate name input

Display their rank accordingly
Repeat until users stop

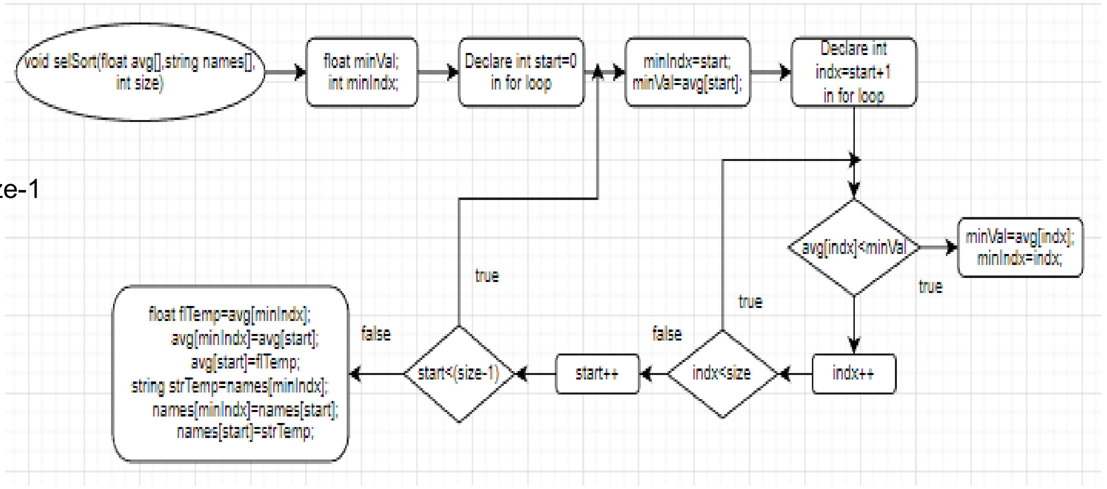


C
cout<<endl<<"Would you like to search for your rank(y or n) : ";
cin>>search;
search!='y'&&search!='n'&&search!='Y'&&search!='N'
true
cout<<"Invalid choice. Try again (y or n): ";
cin>>search;
cout<<"Name: ";
cin>>nmeSrch;
rank=rnkSrch(names,numPlyr,nmeSrch);
rank==-1
true
cout<<"Invalid name."<<endl
false
cout<<nmeSrch<<"'s rank is "<<rank+1<<endl;
cout<<"Again(y or n)? ";
cin>>again;
again!='y'&&again!='n'&&again!='Y'&&again!='N'
cout<<"Invalid choice. Try again (y or n): ";
cin>>again;
true
again!='n'&&again!='N'
cout<<endl<<endl;

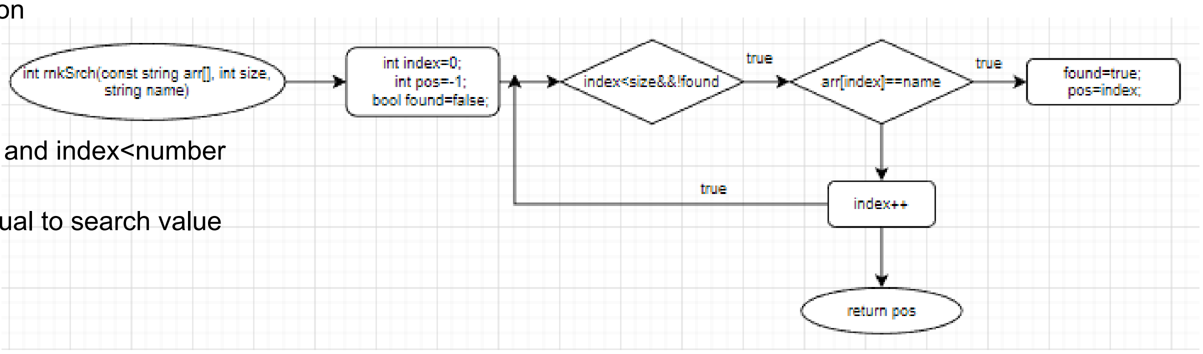
```
-Bubble Sort Function
For maxEle=each subscript in the vector,
from the last to the first
    For index=0 to maxEle-1
      If vector[index]>vector[index+1]
          Swap vector[index] with
                    vector[index+1]
      End if
    End for
End for
```


void bbleSort(vector<int>&num, int size)
Declare int maxEle=szie-1 in for loop
Declare int i=0 in for loop
num[indx]>num[indx+1]
int temp=num[indx]; num[indx]=num[indx+1]; num[indx+1]=temp;
indx++
indx<maxEle
true
false
maxEle--
maxEle>0
true

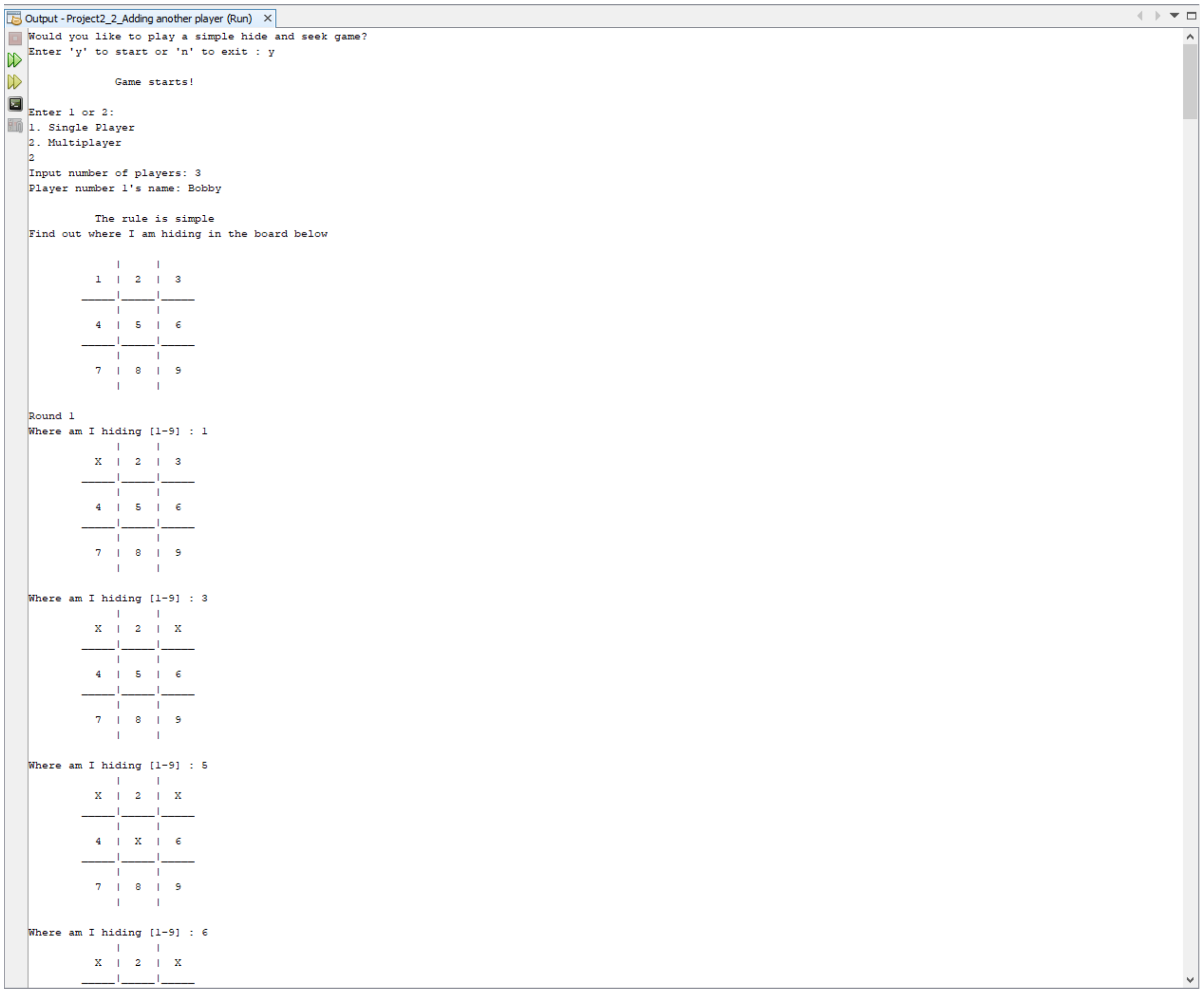
```
-Selection Sort Function
For start=each array avg subscript,
from the first to last-1

    minIndx=start
    minVal=avg[start]

   For index=start+1 to size-1

      If avg[indx]<minVal
         minVal=avg[indx]
         minIndx=indx

      End if
    End For

    Swap avg[minIndx] with avg[start]
    Swap names[minIndx] with names[start]
End For
```


void selSort(float avg[],string names[], int size)
float minVal; int minIndx;
Declare int start=0 in for loop
minIndx=start; minVal=avg[start];
Declare int indx=start+1 in for loop
avg[indx]<minVal
true
minVal=avg[indx]; minIndx=indx;
indx++
indx<size
true
false
start++
start<(size-1)
true
false
float flTemp=avg[minIndx]; avg[minIndx]=avg[start]; avg[start]=flTemp; string strTemp=names[minIndx]; names[minIndx]=names[start]; names[start]=strTemp;

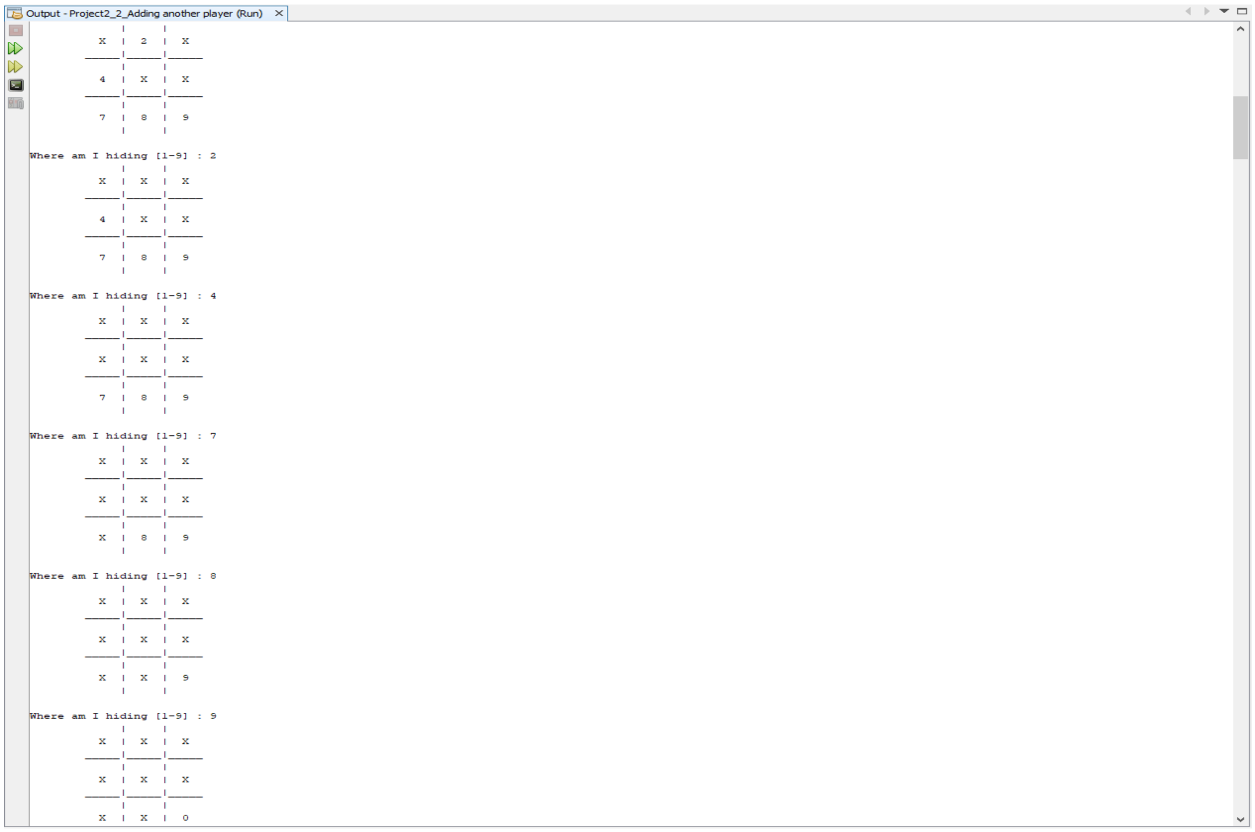
```
-Rank search function
Set found to false
Set Position to -1
Set index to 0
While found is false and index<number
    of elements
 If array[index] is equal to search value
       found=true
       position=index
 End If
       Increment index
End While
Return position
```


int rnkSrch(const string arr[], int size, string name)
int index=0; int pos=-1; bool found=false;
index<size&&!found
true
arr[index]==name
true
found=true; pos=index;
true
index++
return pos

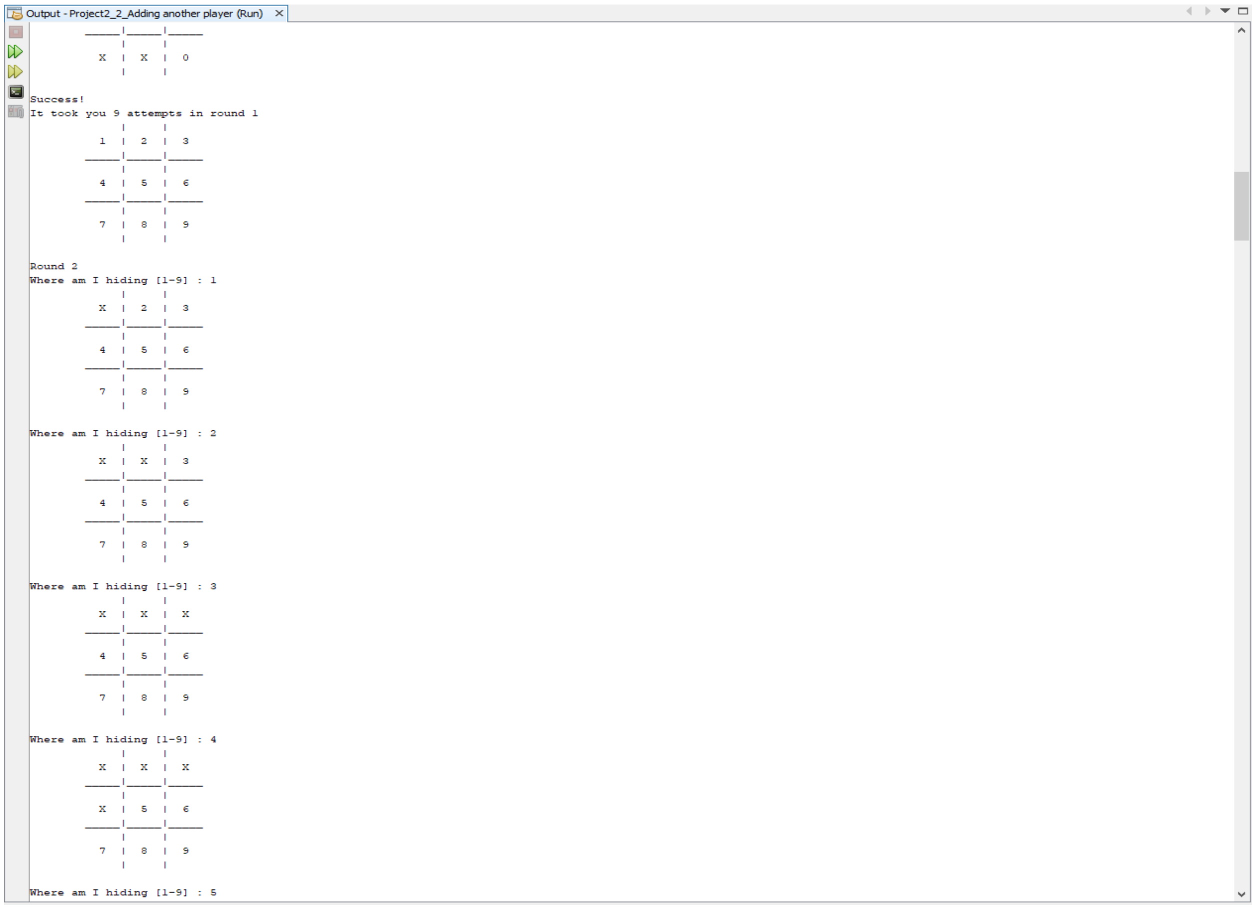
Proof of a working Program



```
Would you like to play a simple hide and seek game?
Enter 'y' to start or 'n' to exit : y

             Game starts!

Enter 1 or 2:
1. Single Player
2. Multiplayer
2
Input number of players: 3
Player number 1's name: Bobby

          The rule is simple
Find out where I am hiding in the board below

             |     |
          1  |  2  |  3
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Round 1
Where am I hiding [1-9] : 1
             |     |
          X  |  2  |  3
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 3
             |     |
          X  |  2  |  X
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 5
             |     |
          X  |  2  |  X
        _____|_____|_____
             |     |
          4  |  X  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 6
             |     |
          X  |  2  |  X
        _____|_____|_____
```

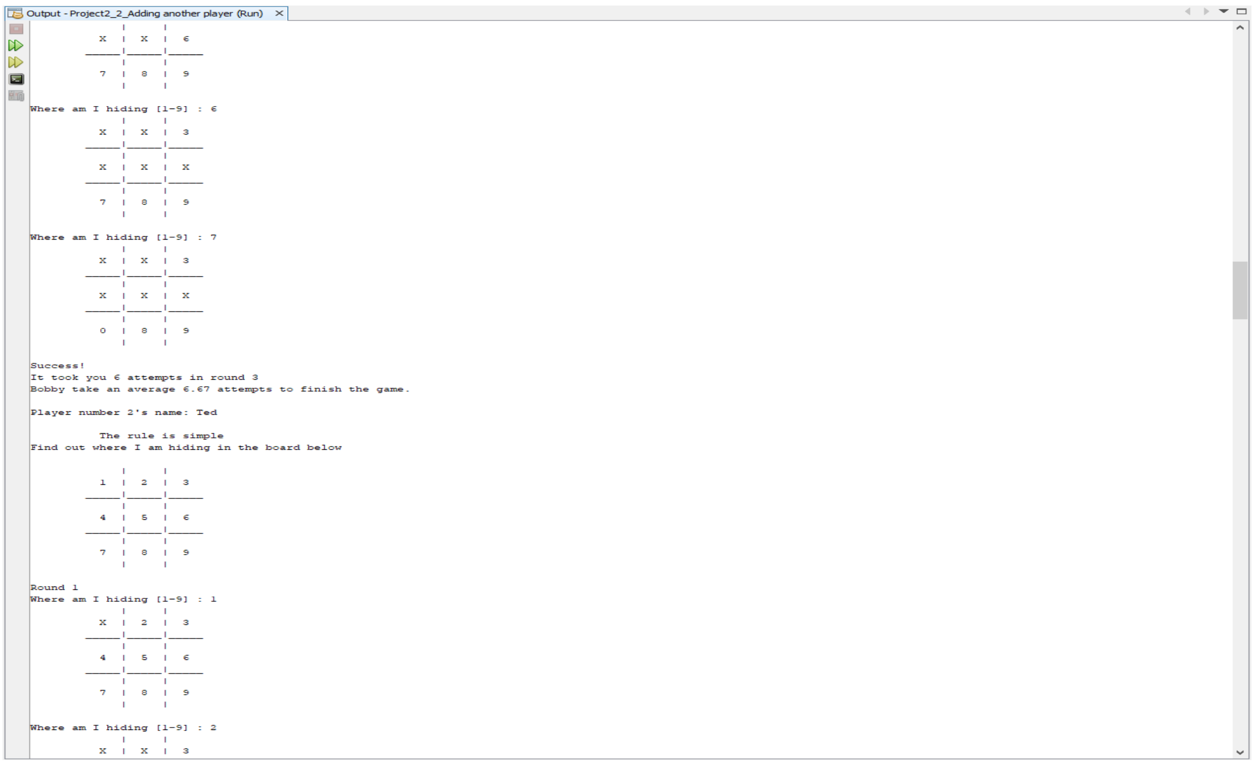

Output - Project2_2_Adding another player (Run)

```
              |     |
           X  |  2  |  X
        ______|_____|______
              |     |
           4  |  X  |  X
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 2
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           4  |  X  |  X
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 4
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 7
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  8  |  9
              |     |

Where am I hiding [1-9] : 8
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  9
              |     |

Where am I hiding [1-9] : 9
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  X  |  O
```


Output - Project2_2_Adding another player (Run)

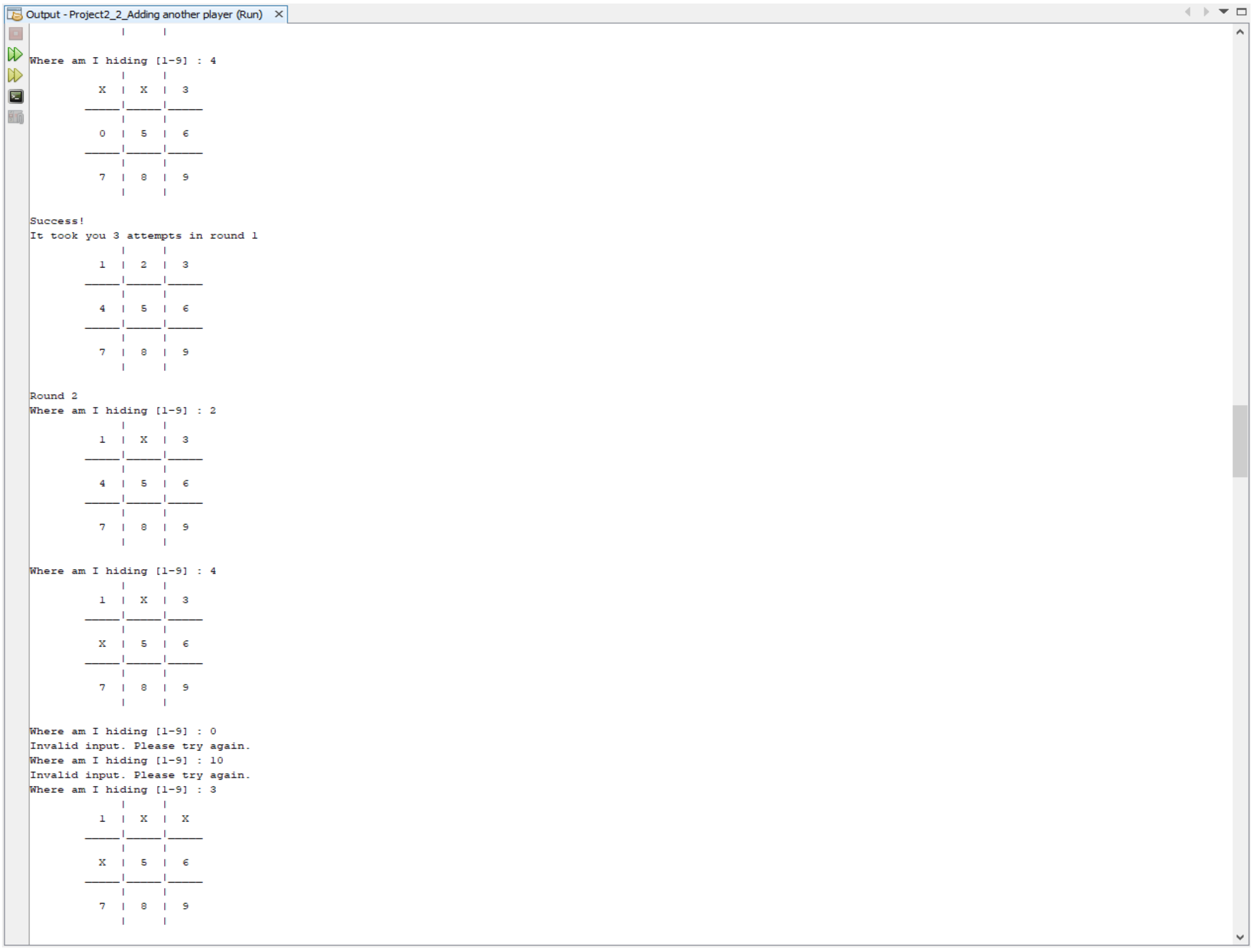
```
        ______|_____|______
              |     |
           X  |  X  |  O
              |     |

Success!
It took you 9 attempts in round 1
              |     |
           1  |  2  |  3
        ______|_____|______
              |     |
           4  |  5  |  6
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Round 2
Where am I hiding [1-9] : 1
              |     |
           X  |  2  |  3
        ______|_____|______
              |     |
           4  |  5  |  6
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 2
              |     |
           X  |  X  |  3
        ______|_____|______
              |     |
           4  |  5  |  6
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 3
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           4  |  5  |  6
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 4
              |     |
           X  |  X  |  X
        ______|_____|______
              |     |
           X  |  5  |  6
        ______|_____|______
              |     |
           7  |  8  |  9
              |     |

Where am I hiding [1-9] : 5
```

12



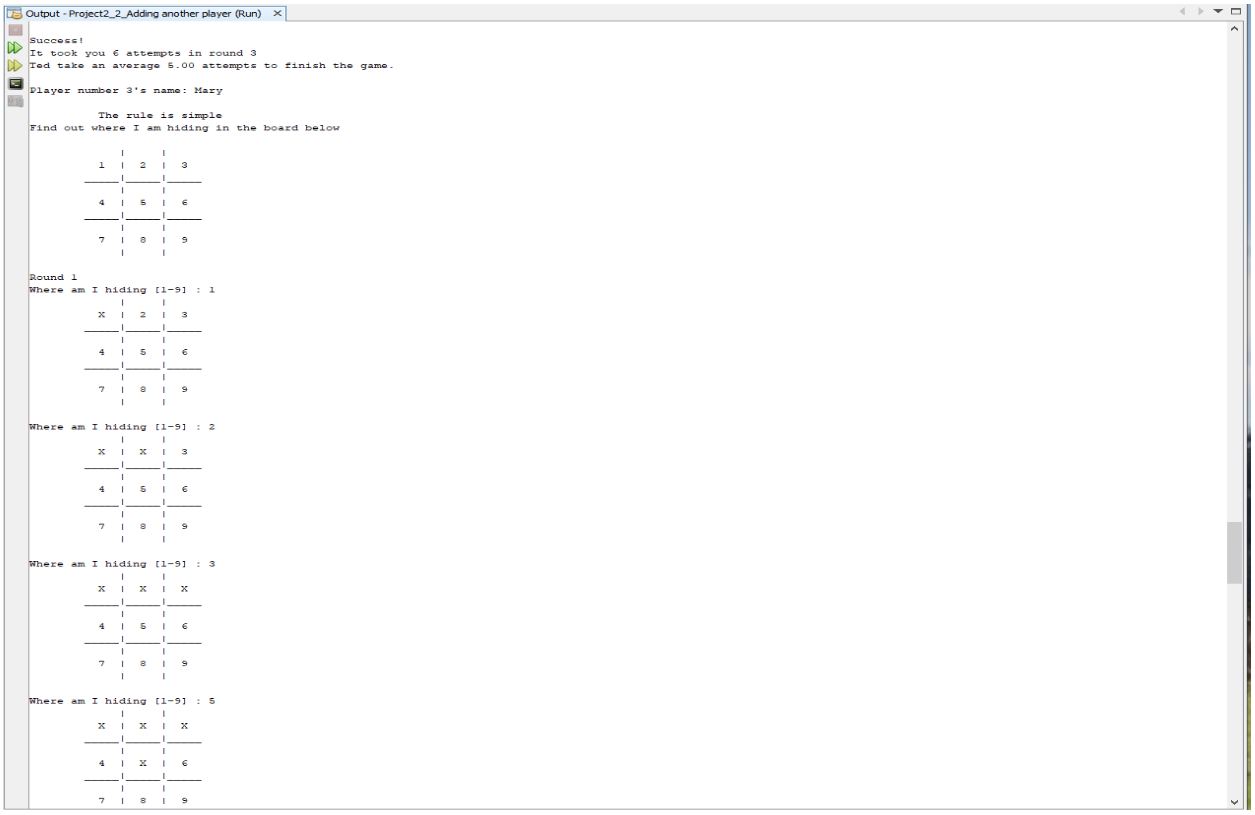
Output - Project2_2_Adding another player (Run)

```
          |     |
      X   |  X  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Where am I hiding [1-9] : 6
          |     |
      X   |  X  |  3
     _____|_____|_____
          |     |
      X   |  X  |  X
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Where am I hiding [1-9] : 7
          |     |
      X   |  X  |  3
     _____|_____|_____
          |     |
      X   |  X  |  X
     _____|_____|_____
          |     |
      O   |  8  |  9
          |     |

Success!
It took you 6 attempts in round 3
Bobby take an average 6.67 attempts to finish the game.

Player number 2's name: Ted

          The rule is simple
Find out where I am hiding in the board below

          |     |
      1   |  2  |  3
     _____|_____|_____
          |     |
      4   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Round 1
Where am I hiding [1-9] : 1
          |     |
      X   |  2  |  3
     _____|_____|_____
          |     |
      4   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Where am I hiding [1-9] : 2
          |     |
      X   |  X  |  3
```



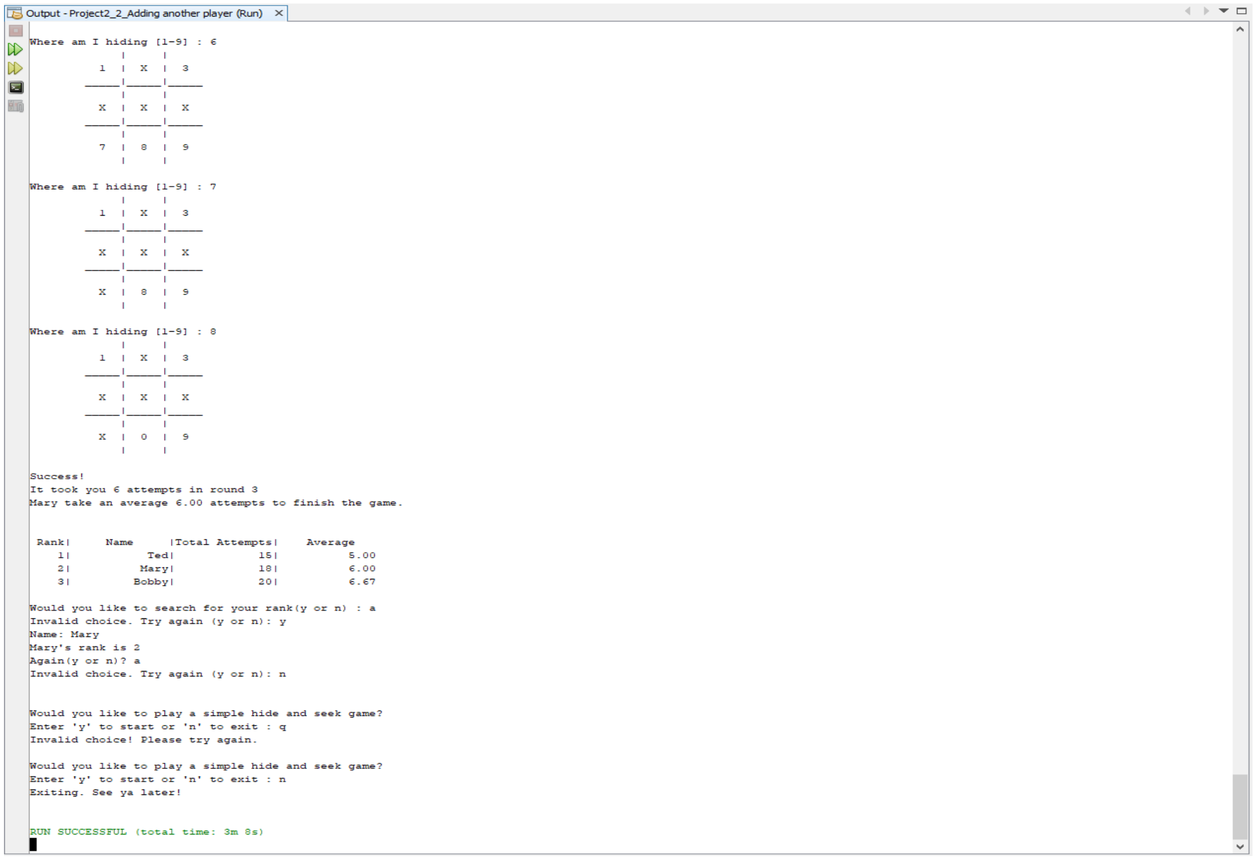
Output - Project2_2_Adding another player (Run)

```
          |     |

Where am I hiding [1-9] : 4
          |     |
      X   |  X  |  3
     _____|_____|_____
          |     |
      O   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Success!
It took you 3 attempts in round 1
          |     |
      1   |  2  |  3
     _____|_____|_____
          |     |
      4   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Round 2
Where am I hiding [1-9] : 2
          |     |
      1   |  X  |  3
     _____|_____|_____
          |     |
      4   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Where am I hiding [1-9] : 4
          |     |
      1   |  X  |  3
     _____|_____|_____
          |     |
      X   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |

Where am I hiding [1-9] : 0
Invalid input. Please try again.
Where am I hiding [1-9] : 10
Invalid input. Please try again.
Where am I hiding [1-9] : 3
          |     |
      1   |  X  |  X
     _____|_____|_____
          |     |
      X   |  5  |  6
     _____|_____|_____
          |     |
      7   |  8  |  9
          |     |
```



Output - Project2_2_Adding another player (Run)

```
Success!
It took you 6 attempts in round 3
Ted take an average 5.00 attempts to finish the game.

Player number 3's name: Mary

          The rule is simple
Find out where I am hiding in the board below

             |     |
          1  |  2  |  3
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Round 1
Where am I hiding [1-9] : 1
             |     |
          X  |  2  |  3
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 2
             |     |
          X  |  X  |  3
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 3
             |     |
          X  |  X  |  X
        _____|_____|_____
             |     |
          4  |  5  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 5
             |     |
          X  |  X  |  X
        _____|_____|_____
             |     |
          4  |  X  |  6
        _____|_____|_____
             |     |
          7  |  8  |  9
```



Output - Project2_2_Adding another player (Run)

```
Where am I hiding [1-9] : 6
             |     |
          1  |  X  |  3
        _____|_____|_____
             |     |
          X  |  X  |  X
        _____|_____|_____
             |     |
          7  |  8  |  9
             |     |

Where am I hiding [1-9] : 7
             |     |
          1  |  X  |  3
        _____|_____|_____
             |     |
          X  |  X  |  X
        _____|_____|_____
             |     |
          X  |  8  |  9
             |     |

Where am I hiding [1-9] : 8
             |     |
          1  |  X  |  3
        _____|_____|_____
             |     |
          X  |  X  |  X
        _____|_____|_____
             |     |
          X  |  0  |  9
             |     |

Success!
It took you 6 attempts in round 3
Mary take an average 6.00 attempts to finish the game.


 Rank|     Name      |Total Attempts|    Average
    1|            Ted|            15|          5.00
    2|           Mary|            18|          6.00
    3|          Bobby|            20|          6.67

Would you like to search for your rank(y or n) : a
Invalid choice. Try again (y or n): y
Name: Mary
Mary's rank is 2
Again(y or n)? a
Invalid choice. Try again (y or n): n


Would you like to play a simple hide and seek game?
Enter 'y' to start or 'n' to exit : q
Invalid choice! Please try again.

Would you like to play a simple hide and seek game?
Enter 'y' to start or 'n' to exit : n
Exiting. See ya later!


RUN SUCCESSFUL (total time: 3m 8s)
```

Program coding:

```
/*
 * File:   main.cpp
 * Author: Triet Huynh
 * Created on July 30th, 2022, 10:30 AM
 * Purpose: Project 1_Simple Hide and Seek game in a 9 slots board for 1 player
 */

//System Libraries
#include <iostream>  //I/O Library
#include <cstdlib>   //Random Function Library
#include <ctime>     //Time Library
#include <iomanip>   //Formatting Library
#include <cmath>     //math library
#include <fstream>   //file stream
#include <vector>
using namespace std;

//User Libraries

//Global Constants, no Global Variables are allowed
//Math/Physics/Conversions/Higher Dimensions - i.e. PI, e, etc...

//Function Prototypes
void dplTble(const char [][3]);
float playGme(string,int&);
void playGme(int);
bool win(char[][3],int);
void bbleSort(vector<int>&, int);
void selSort(float [],string[], int);
int rnkSrch(const string [], int, string);

//Execution Begins Here!
int main(int argc, char** argv) {
    //Set the random number seed
    srand(static_cast<unsigned int>(time(0)));

    //Declare Variables
    const unsigned short ROWS=3,
            COLS=3;
    int gmeType,        //single player or multiplayer
        numPlyer,       //number of players, 1 or 2
            atmpSum;    //Sum of attempts that player takes to finish the game
    char start;         //user choice to start or quit the game
    float atmpAvg;    //average number of attempts it takes for player to succeed per round
    string plyer;       //player's name
    bool win;           //true if player guess correctly, false otherwise
    //Initialize or input i.e. set variable values
    do{
        cout<<"Would you like to play a simple hide and seek game?"<<endl;
        cout<<"Enter 'y' to start or 'n' to exit : ";
        cin>>start;
    if(start=='y'||start=='Y'){
        cout<<endl<<"            Game starts!  "<<endl<<endl;
        cout<<"Enter 1 or 2:"<<endl
                <<"1. Single Player"<<endl
                <<"2. Multiplayer"<<endl;
        cin>>gmeType;
        //Input validation
        while(gmeType!=1&&gmeType!=2){
            cout<<"Invalid choice. Enter [1-2]."<<endl;
            cin>>gmeType;
        }
        if(gmeType==1){
            cout<<"Enter player's name: ";
            cin>>plyer;
            playGme(plyer,atmpSum);
        }else if(gmeType==2){
            cout<<"Input number of players: ";
            cin>>numPlyer;
            playGme(numPlyer);
        }
    }else if(start=='n'||start=='N'){   //quits game
        cout<<"Exiting. See ya later!"<<endl<<endl;
        exit(0);
    }else{                              //invalid input
        cout<<"Invalid choice! Please try again."<<endl<<endl;
    }
    }while(start!='n'&&start!='N');
    //Exit stage right or left!
    return 0;
}
//Function to display game table
void dplTble(const char table[][3]){
    cout<<"\t     |     |     "<<endl;
    cout<<"\t  "<<table[0][0]<<"  |  "<<table[0][1]<<"  |  "<<table[0][2]<<"  "<<endl;
    cout<<"\t_____|_____|_____"<<endl;
    cout<<"\t     |     |     "<<endl;
    cout<<"\t  "<<table[1][0]<<"  |  "<<table[1][1]<<"  |  "<<table[1][2]<<"  "<<endl;
    cout<<"\t_____|_____|_____"<<endl;
    cout<<"\t     |     |     "<<endl;
    cout<<"\t  "<<table[2][0]<<"  |  "<<table[2][1]<<"  |  "<<table[2][2]<<"  "<<endl;
    cout<<"\t     |     |     "<<endl<<endl;
}
//Single player game
float playGme(string plyer,int &atmpSum){
```

```
const int ROWS=3,   //game board has 3 rows and 3 columns
        COLS=3;
int choice,          //player's guess
        hide,        //where the object is hiding
        attmps,      //number of attempts it took for user to find the right spot
        temp,        //temporary value to hold number of attempts in each round
        rounds;      //3 rounds in total
float atmpAvg;    //average number of attempts it takes for player to succeed per round
    cout<<endl<<"          The rule is simple"<<endl;
    cout<<"Find out where I am hiding in the board below"<<endl<<endl;
    ofstream outputFile(plyer);
    atmpSum=0;
    //start game from round 1->3
    for(rounds=0;rounds<3;rounds++){
        char table[ROWS][COLS]={{'1','2','3'},{'4','5','6'},{'7','8','9'}};   //game's board
        hide=(rand()%9)+1; //generate a random number as hiding slot
        attmps=0,temp=0;
        //display table
        dplTble(table);
        cout<<"Round "<<rounds+1<<endl;
        do{
            cout<<"Where am I hiding [1-9] : ";
            cin>>choice;
            //input validation
            if(choice<1||choice>9)
                cout<<"Invalid input. Please try again."<<endl;
            else{
            attmps++;
            switch (choice){
                case 1:{
                    if(choice==hide)
                    {
                        table[0][0]='0';
                        //display updated table
                        dplTble(table);
                        break;
                    }else
                    {
                        table[0][0]='X';
                        //display updated table
                        dplTble(table);
                        break;
                    }
                }
            case 2:{
                if(choice==hide)
                    {
                        table[0][1]='0';
                        dplTble(table);
            break;
        }else
        {
            table[0][1]='X';
            dplTble(table);
            break;
        }
    }
case 3:{
    if(choice==hide)
        {
            table[0][2]='0';
            dplTble(table);
            break;
        }else
        {
            table[0][2]='X';
            dplTble(table);
            break;
        }
}
case 4:{
    if(choice==hide)
        {
        table[1][0]='0';
            dplTble(table);
            break;
        }else
        {
            table[1][0]='X';
            dplTble(table);
            break;
        }
}
case 5:{
    if(choice==hide)
        {
            table[1][1]='0';
            dplTble(table);
            break;
        }else
        {
            table[1][1]='X';
            dplTble(table);
            break;
        }
}
case 6:{
    if(choice==hide)
```

```
        {
            table[1][2]='0';
            dplTble(table);
            break;
        }else
        {
            table[1][2]='X';
            dplTble(table);
            break;
        }
}
case 7:{
    if(choice==hide)
        {
            table[2][0]='0';
            dplTble(table);
            break;
        }else
        {
            table[2][0]='X';
            dplTble(table);
            break;
        }
}
case 8:{
    if(choice==hide)
        {
            table[2][1]='0';
            dplTble(table);
            break;
        }else
        {
            table[2][1]='X';
            dplTble(table);
            break;
        }
}
case 9:{
    if(choice==hide)
        {
            table[2][2]='0';
            dplTble(table);
            break;
        }else
        {
            table[2][2]='X';
            dplTble(table);
            break;
        }
```

```
                }
                }
                }
            }while(choice!=hide);
            if(win(table,COLS)==true){
            cout<<"Success!"<<endl;
            cout<<"It took you "<<attmps<<" attempts in round "<<rounds+1<<endl;
            }
            outputFile<<attmps<<endl;   //save player's attempts in file
        }
        outputFile.close();
        ifstream inputFile;
        inputFile.open(plyer);  //create a file with player's name and store attempts for record
        while(inputFile>>temp){
            atmpSum+=temp;
        }
        inputFile.close();
        atmpAvg=abs(static_cast<float>(atmpSum)/3.0);
        cout<<fixed<<showpoint<<setprecision(2);
        cout<<plyer<<" take an average "<<atmpAvg<<" attempts to finish the game."<<endl<<endl;
        return atmpAvg;
}
//check for winning status
bool win(char table[][3],int cols){
    for(int i=0;i<3;i++){
        for(int j=0;j<cols;j++){
            if(table[i][j]=='0')
                return true;
        }
    }
    return false;
}
//Multiplayer game
void playGme(int numPlyr=1){
    string plyer,           //player's name
            nmeSrch;        //player's name to search for rank
    int atmpSum;            //sum of player attempts after 3 rounds
    float atmpAvg;          //average attempts to finish 3 rounds
    string names[numPlyr];  //players ' names array
    float avg[numPlyr];     //and their average attempts
    vector<int>sum;         //player's attempts total array
    char search,            //whether player wants to search for their rank
            again;          //search again? yes or no
    int rank;               //player's rank
    //perform game until last player is done and fill in names, total score and avg score
    for(int i=0;i<numPlyr;i++){
        cout<<"Player number "<<i+1<<"'s name: ";
        cin>>plyer;
        names[i]=plyer;
        atmpAvg=playGme(plyer,atmpSum);
        avg[i]=atmpAvg;
        sum.push_back(atmpSum);
    }
    //Sort the arrays and vectors
    bbleSort(sum,sum.size());
    selSort(avg,names,numPlyr);
    //Display the ranking board
    cout<<endl<<" Rank|     Name     |Total Attempts|    Average   "<<endl;
    for(int i=0;i<numPlyr;i++){
        cout<<setw(5)<<i+1<<"|"<<setw(14)<<names[i]<<"|"
                <<setw(14)<<sum[i]<<"|"
                <<setw(14)<<avg[i]<<endl;
    }
    //Prompt user for rank search
    cout<<endl<<"Would you like to search for your rank(y or n) : ";
    cin>>search;
    //Input validation
    while(search!='y'&&search!='n'&&search!='Y'&&search!='N'){
        cout<<"Invalid choice. Try again (y or n): ";
        cin>>search;
    }
    do{
        cout<<"Name: ";
        cin>>nmeSrch;
        rank=rnkSrch(names,numPlyr,nmeSrch);
        if(rank==-1)
            cout<<"Invalid name."<<endl;
        else
            cout<<nmeSrch<<"'s rank is "<<rank+1<<endl;
        cout<<"Again(y or n)? ";
        cin>>again;
        //Input validation
        while(again!='y'&&again!='n'&&again!='Y'&&again!='N'){
            cout<<"Invalid choice. Try again (y or n): ";
            cin>>again;
        }
    }while(again!='n'&&again!='N'); //repeat until player enter no to repeat
    cout<<endl<<endl;
}
//Bubble sort on a vector
void bbleSort(vector<int>&num, int size){
    for(int maxEle=size-1;maxEle>0;maxEle--){
        for(int indx=0;indx<maxEle;indx++){
            //if first value is greater than second value, swap them
            if(num[indx]>num[indx+1]){
                int temp=num[indx];
                num[indx]=num[indx+1];
                num[indx+1]=temp;
            }
        }
    }
}
//Dual sort with array names and array average using selection sort
void selSort(float avg[],string names[], int size){
    float minVal;
    int minIndx;
    for(int start=0;start<(size-1);start++){
        minIndx=start;
        minVal=avg[start];
        for(int indx=start+1;indx<size;indx++){
            if(avg[indx]<minVal){
                minVal=avg[indx];
                minIndx=indx;
            }
        }
        //if value of index indx in first array is less than min value, swap them
        float flTemp=avg[minIndx];
        avg[minIndx]=avg[start];
        avg[start]=flTemp;
        //swap elements in second array accordingly
        string strTemp=names[minIndx];
        names[minIndx]=names[start];
        names[start]=strTemp;
    }
}
//search for player's name to find out their rank using linear search
int rnkSrch(const string arr[], int size, string name){
    int index=0;
    int pos=-1;
    bool found=false;
    while(index<size&&!found){
        if(arr[index]==name)
        {
            found=true;
            pos=index;
        }
        index++;
    }
    return pos;
}
```