Mastermind jr.

Project1

CSC- 5 – 46091 Intro C++

Jonathan Balisky

21 – July – 2015

Content

1. Introduction

Rules and Gameplay

This mastermind jr. is a simpler version of the original mastermind aimed at younger children. The original mastermind starts by randomly generating four different colors in a row. Then without showing it to the player the player has to find out that pattern of colors with a certain amount of guesses. Each time after the player guesses, it will tell the player how many colors are correct and are at the right place, and how many colors are the right colors but not in the right place. Due to my simple knowledge of c++ of change the colors to numbers (0-9). The program will generate 3 random numbers and the player has 9 chances to guess to number. If the player fails to guess the right number in 9 tries, the program will end and ask the player if they would like to try again.

Thoughts after Program

If I would to do this game again I would certainly like a graphical interface. For me much of the involvement in games comes due not only to mental stimulation but also visual stimulation.

The next step would be to use colors instead of numbers and have the screen populate with the guessed after each guess. If might also be nice to some type of animation if the player wins or loses. I also realize that I do not display the correct answer after the player losses. This can create some frustration for the player and should be corrected in the next version

2. Development

Approach Strategy

When searching for a project I wanted something fairly simply but still fun to play. That's why I came up with mastermind jr. it's still requires significant logic but even and simpleton like me can play.

There were several key elements that I needed for this project. First was the srand function. If the game had the same problem each time, it's quite clear that not only would the game be very easy it also would be boring. Using this srand function mod 10 allowed me to create a random problem for the user to solve each time. Next huge need was arrays. Using arrays allowed me to do two things. First to be able input a number in which each character or integer is stored in its own separate "box". This allowed me to then be able to check compare two arrays, or the user guess vs the answer, "box" by "box". Without this there would be many more lines of code…which would be against "Marks law". The last main component was the "for" loops. The "for" loops where essential, using such simple logic such as i = 0, if i<3, go ahead and do what comes next, and after your done add one to i. Combining this with arrays and the random function allowed me easy access to the answers. With the arrays and "for" loop I was able to create the bulk of the logic in this game.

3. Research

I. Arrays
This is best option to compare each digit of the answer each time the player enters their guess. It was possible to use if and else-ifs to do this, but it would require much more lines of code aka a waste of time and space. This also allow me to later expand the game without too much difficulty.

II. Parallel arrays
In this project because I used arrays I then could use parallel arrays. I needed parallel arrays to check in a particular location (answer[0] compare to usrG[0]), was the same or not. Then after comparing on location I could then loop the next (answer[0] compare to usrG[1]).

III. Ascii to integer
For ease of input I used a string as the user's guess. Because string is stored as a character and an asci character cannot be compared directly to and integer there needed to be some type of conversion. Using a little concept that I found in my homework with a "for" loop and an array simply subtract 48 from each value and it can be compared to an integer

IV. String.length
Because you can never be sure what the user might input I needed some type of check for the user's guess. String.length was used check the length of the digits entered so if the user entered the wrong amount of digits I could loop and ask then to try again.

4. Variables list

Type	**Variable Name**	**Description**	**Line**
int	count		31
	realO[4]	the generated number save	32
	real[4]	the generated number compare to guess	33
	X	right place counter	35
	O	right number counter	35
	garr[4]	guess array after check validation	159
char	choice	choose which level	29

	display	choose whether display the correct number or not	30, 145
string	sg	guess number input to check validation	157
bool	repeat	check whether generated digits are the same	34
	valid	check sg validation	158
	same[4]	to remember the place when the digit is at the right and avoid to count with int O	160
time_t	tstart	record start time	230
	tend	record end time	230
ofstream	output		36

5. Topic Covered (Checklist)

Chapter	type	code	line
2.1 Variables	int	int xs	37
2.2 Input Output	Getline	Getlinw(cin,useG)	75
	cout	cout<<"Input your guess: "<<endl;	74
	endl	cout<<endl;	55
2.3 data types	Short	Short ox	38
	bool	bool repeat = false;	34
	string	string sg;	157
2.4 condition	=	Int i=0	62
	==	while(isvalid(usrG) == false);	78
	++	i++;	62
		//variables	

2.5 style	comment		29
3.1 boolean expression	!=	if(usrG.length() != 3){	139
	<, >, ||	if (usrG[i] <48 || usrG[i] > 57){	145
3.2 multiway branches	if	if(usrG.length() != 3){	139
	else	else{	52
	nested	do{	42
		if(input.is_open()){	45
		for (int i = 0; i<11; i++){	46
3.3 type of loop	for	for(int i=0;i<11;i++){	46
	do-while	do{ }while(usrG[0]=='Y');	45,103
4.2 predefined function	srand, time	srand(time(0));	41
	rand	answer[i] = rand()%10;	121
4.3function prototypes	Int	int compare(int [], short&, string, bool[]);	21
	Bool	bool isvalid(string);	20
5.1 void function	void	void prepare(int [], int&, short&, int&, bool[]);	19
5.2 call-by-reference	&	int compare(int [], short&, string, bool[]);	21
6.1 streams and basic	ofsream declare	ofstream output;	34
	Ifstream declare	Ifstream input	35
	Input	input.open("instructions.txt");	44
	Input.close	input.close();	50
	output	output.open("Answer.txt");	60
	close	output.close();	65
7.1 array	int array	Int answer[SIZE]	31
	bool aray	Bool mathc[SIZE]	32

6. Libraries included

- <cstdlib>
- <iostream>
- <ctime>
- <fstream>

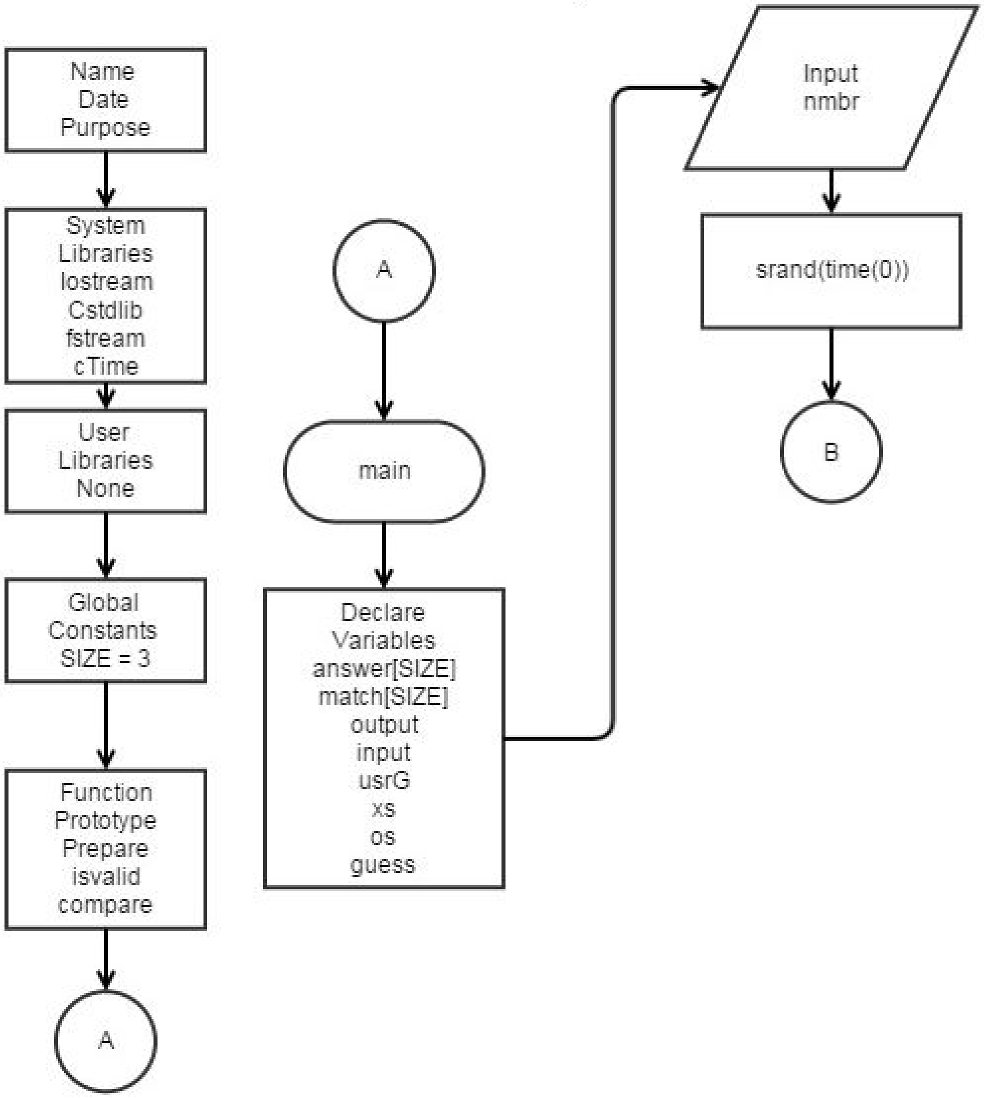
7. Pseudo code

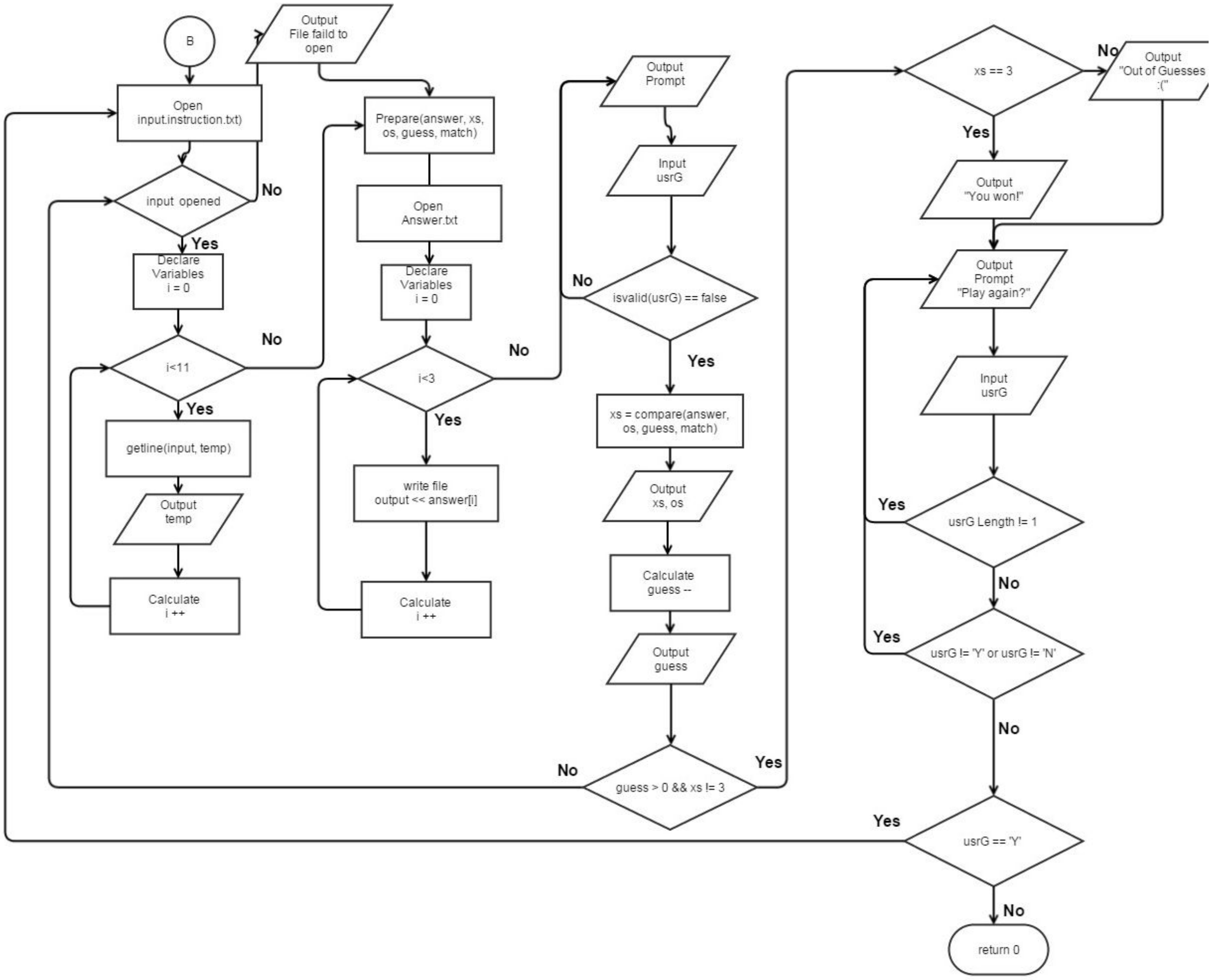
```
Set time seed

Output instruction from file

Do

        Call prepare function

                Random answer and initialize other variables
```

```
    Output the answer to file
    Game start
    do
        Input guess number
        Call isvalid to check validation
        Call compare function
        Display Xs and Ox (result)
        Guess -1
    While guess>0 and guess answer is not correct
    If x==3 output win
    Else output lose
    Ask for another new game
While(Yes)
```

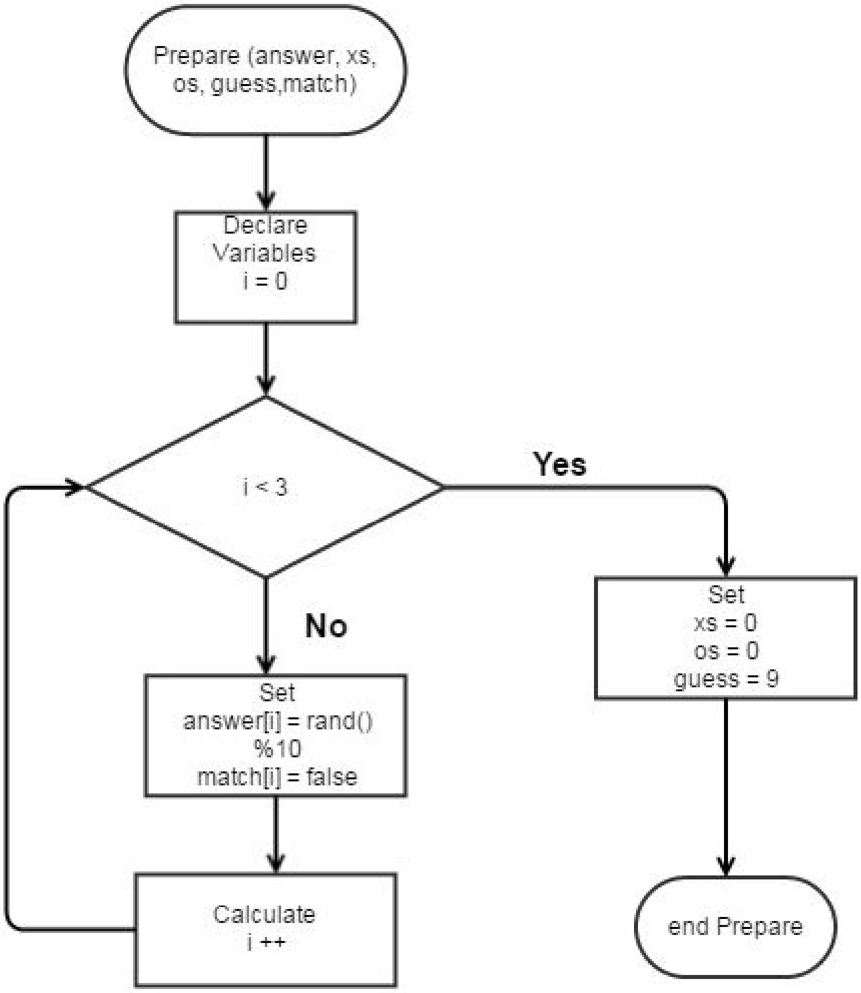
7. Flowchart

Midterm prob 1

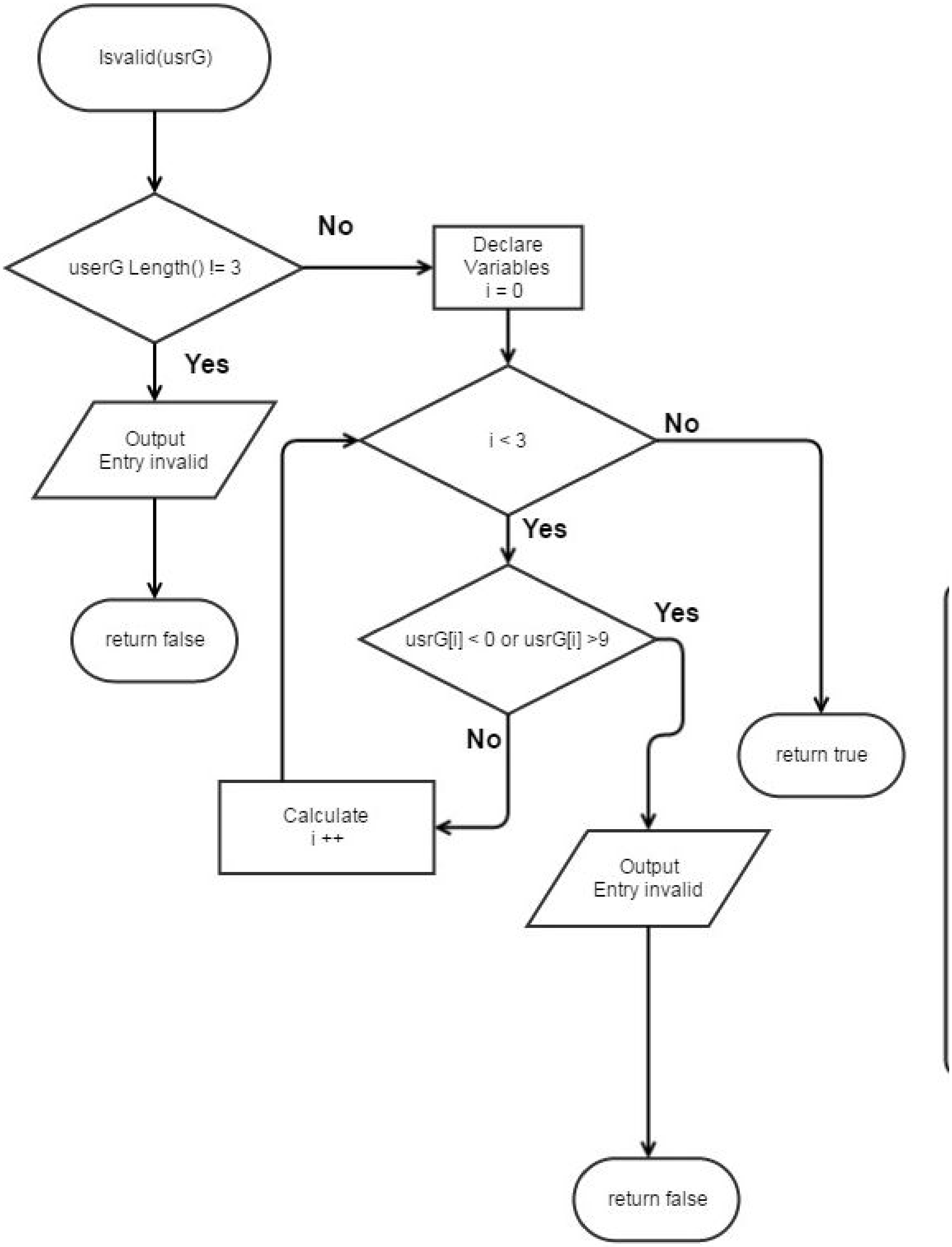



B
Open input.instruction.txt)
input opened
No
Output File faild to open
Yes
Declare Variables i = 0
i<11
No
Yes
getline(input, temp)
Output temp
Calculate i ++
Prepare(answer, xs, os, guess, match)
Open Answer.txt
Declare Variables i = 0
i<3
No
Yes
write file output << answer[i]
Calculate i ++
Output Prompt
Input usrG
isvalid(usrG) == false
No
Yes
xs = compare(answer, os, guess, match)
Output xs, os
Calculate guess --
Output guess
guess > 0 && xs != 3
No
Yes
xs == 3
No
Output "Out of Guesses :("
Yes
Output "You won!"
Output Prompt "Play again?"
Input usrG
usrG Length != 1
Yes
No
usrG != 'Y' or usrG != 'N'
Yes
No
usrG == 'Y'
Yes
No
return 0

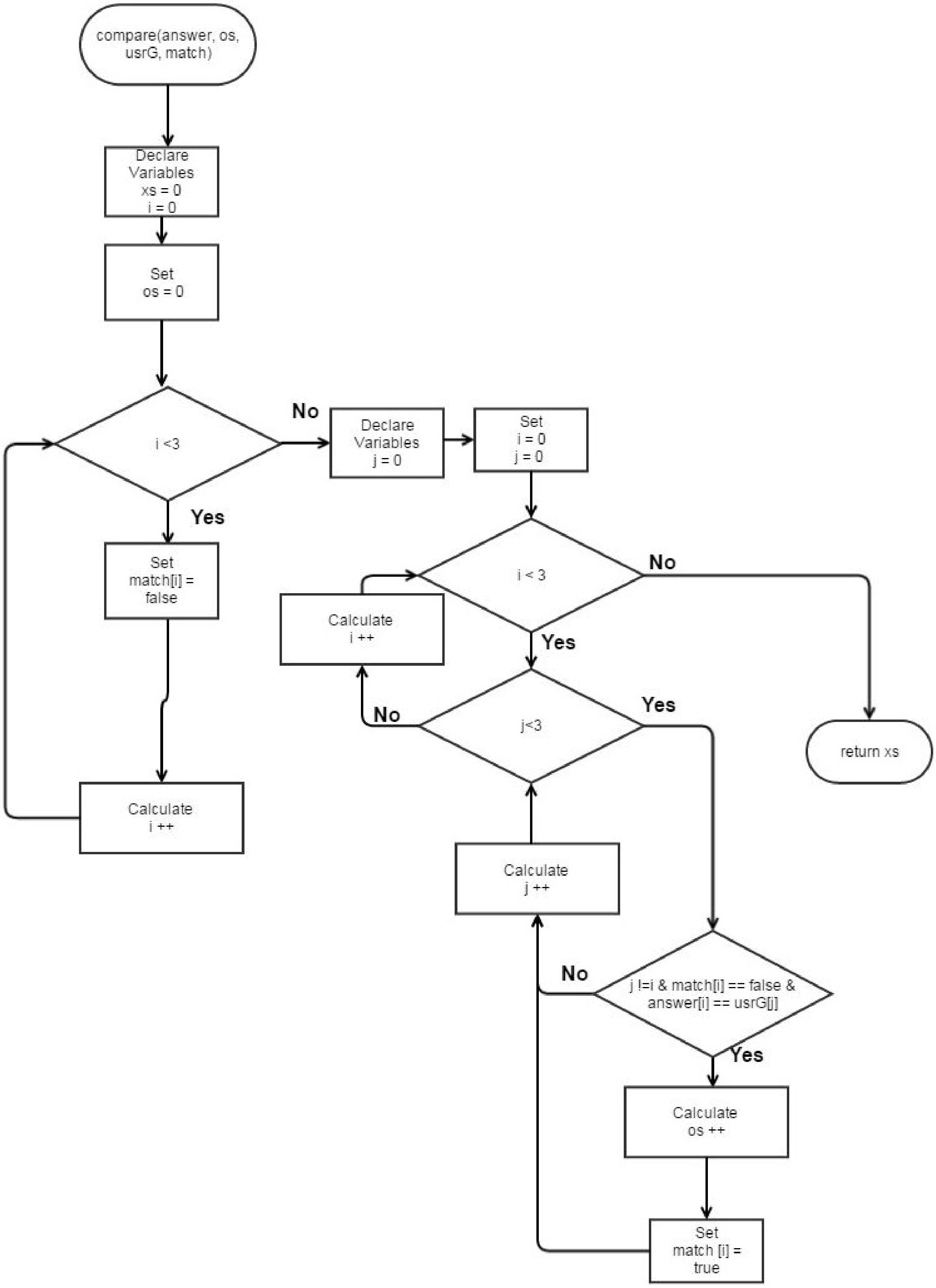
Prepare


Prepare (answer, xs, os, guess,match)
Declare Variables i = 0
i < 3
Yes
No
Set xs = 0 os = 0 guess = 9
Set answer[i] = rand() %10 match[i] = false
Calculate i ++
end Prepare

isValid


Isvalid(usrG)
userG Length() != 3
No
Yes
Declare
Variables
i = 0
Output
Entry invalid
return false
i < 3
No
Yes
usrG[i] < 0 or usrG[i] >9
Yes
No
return true
Calculate
i ++
Output
Entry invalid
return false

isValid


compare(answer, os, usrG, match)
Declare Variables xs = 0 i = 0
Set os = 0
i <3
No
Yes
Set match[i] = false
Calculate i ++
Declare Variables j = 0
Set i = 0 j = 0
i < 3
No
Yes
Calculate i ++
j<3
No
Yes
return xs
Calculate j ++
No
j !=i & match[i] == false & answer[i] == usrG[j]
Yes
Calculate os ++
Set match [i] = true

8. Code

```
/*
 File:   Project 1.cpp
 Author: Jonathan Balisky
 Created on July 19, 2015, 9:18 pM
 Purpose:  Mastermind jr.
 */

//Libraries


#include <iostream>
#include <string>
#include <cstdlib>
#include <fstream>
#include <ctime>

using namespace std;

void prepare(int [], int&, short&, int&, bool[]);
bool isvalid(string);
int compare(int [], short&, string, bool[]);

//Global

const int SIZE = 3;

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

    //Variables

   int answer[SIZE];    //number of pin
   bool match[SIZE]; //Which numbers are matched
   string temp; //For the file output
   ofstream output;     //
   ifstream input; //inputting to the file
   string usrG; //The users guess or input
   int xs; //How many x's
   short os; //How many o'x
   int guess; //How many guess the user had guessed

   srand(time(0));  //setting time seed
   do{
      // cout<<"Call prepare."<<endl;//For diagonostics
       input.open("instructions.txt");
       if(input.is_open()){
           for (int i = 0; i<11; i++){
           getline(input,temp);
           cout<<temp;
           }
           input.close();
        }
       else{
           cout<<"Instructions failed to open"<<endl;
       }
       cout<<endl;

       prepare(answer, xs, os, guess, match); //Initialize

       // cout<<"Call prepare."<<endl;//For diagonostics
       output.open("Answer.txt");

       for (int i = 0; i<3; i++){
        output<<answer[i];
        }
       output.close();


//      for(int i=0;i<3;i++){
        //cout<<answer[i]; //For diagonostics
//      }

       do{
           do {
               cout<<"Input your guess: "<<endl; //User enter guess
               getline(cin,usrG);
               //cin.ignore();

           }while(isvalid(usrG) == false); //Loop until user enters valid answer

           xs=compare(answer, os, usrG, match);

           cout<<"x="<<xs<<endl; //Right numbers in right space
           cout<<"o="<<os<<endl; //How many Correct number but in the incorrect space
           guess--;
           cout<<"Guesses left: "<<guess<<endl;

       }while(guess > 0 && xs != 3);// User out of guess or has guess correctly

       if ( xs == 3){ //user won
           cout<<"Congrats you won!"<<endl;
       }
       else{ //user lost
           cout<<"Sorry but you ran out of guesses. :( "<<endl;

       }
       do{ //Checking for an input of Y or N
           do{ //Checking for input over 1 char
               cout<<"Play again Y/N?"<<endl;
               getline(cin,usrG);
           }while(usrG.length() != 1);
       }while(usrG[0]!=89 && usrG[0]!=78);
           //}while(usrG[0] > 'Y' || usrG < 'N' || (usrG[0]>'N' && usrG[0]<'Y'));
   }while(usrG[0]=='Y' );

   cout<<"See you again next time!"<<endl;


    return 0;
}
```

```
/***************************************Prepare*******************************************
********
 * Purpose: Initializing values for the game.
 * Input: answer, xs, os, guess, match
 * Output:
 * none

******************************************************************************************
******/
void prepare (int answer[], int &xs, short &os, int &guess, bool match[]){

    for (int i = 0; i<3; i++){
        answer[i] = rand()%10; //creating answer from 0-9
        match[i] = false; //Set all to false
        //cout<<"answer = "<<answer[i]; //For diagonostics
    }
   // cout<<endl;
    xs = 0;
    os = 0;
    guess = 9;
}

/***************************************isvalid*******************************************
********
 * Purpose: To check whether or not the user entered 3 numbers
 * Input: usrG
 * Output: True or false

******************************************************************************************
******/
bool isvalid(string usrG){
   // cout<<"Call isvalid."<<endl; //For diagonostics
    if(usrG.length() != 3){
        cout<<"Please enter THREE numbers"<<endl;
        return false;
    }
    else{
        for(int i = 0; i < 3; i++){
            if (usrG[i] <48 || usrG[i] > 57){
                cout<<"Number not entered"<<endl;
                return false;
            }
        }
        // cout<<"number valid"<<endl;//For diagonostics
        return true;
    }

}
/***************************************Compare*******************************************
********
 * Purpose: To compare the user's guess with the answer and return how many
 * were correct or incorrect.
 * Input: answer, os, usrG, match
 * Output:xs

******************************************************************************************
******/
```

```
int compare(int answer[], short &os, string usrG, bool match[]){
    int xs = 0;
    os = 0;

    for(int i = 0; i<3; i++){
       match[i] = false; //int all values to zero again
   }
    //Checking for correct position
   for(int i = 0; i<3; i++){
       if(answer[i] == (usrG[i]-48)){
           xs++;
           match[i] = true;
       }
   }
    //Checking for os
    for(int i = 0; i<3; i++){ //i is position of answer
        for(int j = 0; j<3; j++){ //j is position of usrG(user guess)
            if(j !=i && match[i] == false && answer[i] == usrG[j]-48){
                //cout<<"increased o"<<endl; //For diagonostics
                os++;
                match[i] = true;
            }
        }
    }
    return xs;
}
```