Jose Morales

CSC-5 Intro C++

Dr. Lehr

08/01/2014

Project 2  
**MASTERMIND**

**ULTIMATE VERSION**

**Table of Contents**

**Table of Contents** ............................................................................................................................. 2

**1- Introduction** ............................................................................................................................. 3

Rules and Gameplay

...........................................................................................................

Thoughts after writing the program

...........................................................................................................

**2- Development** ............................................................................................................................... 4

Initial approach ............................................................................................................................... 5

Contents Covered

…………………………………………………………………………………………… 5-6

**3- Flowchart** …………………………………………………………………………. 6

*See flowcharts in GitHub*

**4- Pseudo-Code**................................................................................................. 7-12

**5- Code**............................................................................................................... 12-25

# 1 - Introduction

## Rules and Gameplay:

This program is an adaptation of the popular board game Mastermind. I tried to remain as faithful to the game as possible. For instance, the purpose of the Mastermind game is to break a “code” which consists of four pegs with different colors (or the same) arranged in certain pattern. In this program the objective will be to break a numerical code consisting of two or four different numbers (the code’s size depends on the player’s choosing). Another similarity between the board game and the program is that to play Mastermind two players are required, in the program the player plays against the system. A basic introduction with instructions is included in the beginning in order for the player to be familiarized with the game. The program includes two levels, an easy level in which the code consists of two numbers and seven opportunities are given to the player, and a hard level in which the code contains four different numbers and the player has twelve opportunities to guess it. Similarly to the board game the program will let the user know when they have entered the correct number in the correct position, or when he has entered the correct number in an incorrect position. If the player guesses the code within the opportunities he wins, if not he loses, in both cases he gets the chance to play again or go back to the menu. A difference that I think makes this version of Mastermind harder to play than the original game is the fact that there are 9 different possibilities per position, while in the board game it varies from six to eight. As an interesting fact a brief history of the board game was included.

## Thoughts after writing the program

Even though it is true it was a lot of work, I found the writing of this project to be an extremely fun and useful task. I enjoyed the challenge of having to write something that required me to use all of what has been taught in the past two weeks, and also the fact that it provided me with an invaluable amount of practice. I tried to keep the game simple and easy to understand.

While writing the second stage of this program, project two, I was able to see how knowing arrays and other things such as functions can help to save time and code lines while getting the same result. It made me appreciate the new things that we learned and want to learn more. I can only imagine how many other things can be done more efficiently by learning other tools. It also helped me to comprehend arrays better along with the usefulness of functions in making a program easy to understand.

# 2 - Development

## Strategy used to program the game:

### Initial approach in second project:

For my second project I had a different approach on how to write it. for example, instead of using char variables for user to input the guesses I used strings which I then converted to arrays, also to generate the random numbers I didn’t use single numbers but arrays. Something that was a little bit tricky was checking the guesses with the random numbers. I was able to do so by doing two things. First keeping a counter and second having if statements that allowed me to change the value of the random number if the guess was correct. The only problem with the latter strategy was that then my original numbers would be ruined after the first guess if it was correct. I was able to solve this by making another array to copy the original numbers and use that second array to check instead of the original one, this way, if there was a correct guess it would only affect the second array in the duration of the loop and then the original values would be re-established by copying the original array to the second.

# CONTENTS COVERED

|  |  |  |
| --- | --- | --- |
| Chapter | type | line |
| 2.1 Variables | int | 34 |
| 2.2 Input Output | cin | 73 |
|  | cout | 72 |
|  | endl | 104 |
| 2.3 data types | char | 33 |
|  | float | 598 |
|  | bool | 195 |
|  | string | 48 |
| 2.4 condition | = | 110 |
|  | == | 131 |
|  | ++ | 134 |
| 2.5 style | Setwidth() | 414 |
|  | Comments | 411 |
| 3.1 boolean expression | >=, &&, <= | 122 |
|  | <, >, || | 127, 309 |
| 3.2 multiway branches | switch | 89 |
|  | if | 447 |
|  | else | 149 |
|  | else if | 272 |
|  | nested | 298 |
|  | Break | 347 |
| 3.3 type of loop | for | 431 |
|  | do-while | 180, 188 |
|  | While | 499 |
| 4.2 predefined function | srand, time | 54 |
|  | Rand | 80 |
| 5.1 void function | Void | 369 |
| Function | Int | 26 |
| 5.2 call-by-reference |  | 595 |
| 6.1 streams and basic | ofstream declare | 599 |
|  | output | 612 |
|  | close | 617 |
| 7.1 array | int one dimensional array | 406 |
| 7.2 array in function | pass 2d array to function | 23 |
| 7.3 sorting | Sorting | 636-658 |
| 7.4 multi-dim array | 2D | 404 |

**LIBRARIES INCLUDED**

* #include <cstdlib>// Random numbers
* #include <iostream>
* #include <ctime>
* #include <iomanip> //Output format Library
* #include <string>//used for strings
* #include <fstream>//to open file

# 3. FLOWCHARTS

\* SIDE NOTE: The flowcharts will be uploaded to github for better visualization.

# 4. Pseudo Code

*/\**

*\* File: main.cpp*

*\* Author: Daniel*

*\**

*\* Created on July 19, 2014, 1:15 PM*

*\*/*

*//System Library*

*//Random numbers*

*//Output format Library*

*//used for strings*

*//to open file*

*//User Libraries*

*//Global Constants*

*//Function Prototype*

*//Function for prog intro*

*//Function for level 3*

*//Function to see random numbers in level 3*

*//F:Random numbers generator level 3*

*//F: see random numbers*

*//F: to compare*

*//F:Wins count*

*//F: Lose count*

*//F: total wins loses output file*

*//Function to sort answers in level 3*

*//Execution Begins Here!*

*//Description of the program*

*//Declare Menu Variables*

*//Declaring variables for game part 1*

*//Each one of this represents the numbers the player will enter, g1 for first number and so on...*

*//Each one of this represents the random numbers that will be generated*

*//Declaring variables for game part 2*

*//Each one of this represents the numbers the player will enter, g1 for first number and so on...*

*//Each one of this represents the random numbers that will be generated*

*//Declare Variables for open file, see history of mastermind*

*//Declare Variables for players name function*

*//Declare Variables for w L functions*

*//Counter common to both games*

*//c2=loop*

*//Random Numbers seed*

*//Have user input name*

*//Loop to perform the game until user wants to exit*

*//Setting options using a switch*

*//Option one menu, level 1 of 2 numbers guess*

*//Do loop while random numbers are the same.. ensures different numbers are generated and passed*

*//Setting the random numbers*

*//This is random number 1*

*//Random number 2*

*//Intro level 2*

*//Calling function to output against which player (comp) the user will play*

*//restarting loop counter*

*//restating c's value*

*//Game loop*

*// this is the count of the loop*

*//taking the random number and making it equal to the variable I will use through the game*

*//User inputs guess*

*// This is the validation of the numbers entered, they must be numbers between 0 and 9*

*//taking the char characters and turning them into numbers... use of ASCII //reset count to 0*

*// This is the validation of the numbers entered*

*//reset count to 0*

*//making sure if two numbers are equal in position and value a counter is started, if the counter hits 2, player wins*

*//If guess is equal to random #, change random number's value and count increases*

*//If the counter reaches 2 it means they guessed both numbers, display "you won" message*

*//Boolean C turns false and the loop ends*

*//Calling function to add win*

*//If they enter an invalid value display message*

*//Notifies the player how many numbers are in the right position*

*//this is to verify each number against the other in case the player*

*//inputed the right number in the wrong position*

*cout<<"You guessed: "<<match2<<" numbers in the wrong position"<<endl;//Display message of how many numbers were guessed in the wrong position*

*//this makes sure that if the player didn't win a try again message is displayed*

*//while the count is not greater than 7 or c is false the loop will run*

*//if c is false player lost, display message*

*//If player losses call the lose count function*

*//Give option to play again*

*//User enters Y to play again and N to exit and go back to menu*

*//if the player wants to play again y must be entered, everything else will be interpreted as if //he doesn't want to play and the loop will end and he will be taken back to the menu*

*//End option 1*

*//Begin option 2*

*//game loop*

*//Validation loop so that no 2 numbers are equal*

*//Setting the random numbers*

*//Random numbers 1-4*

*//close validation loop*

*//Intro option 2, 4 random numbers*

*//Calling function to see name of comp "player's name"....*

*//restarting loop count*

*//restating b's value as a bool*

*//Game loop*

*// this is the count of the loop*

*//Savind random numebers memory by assigning them to a new variable*

*//User enters guess;*

*// This is the validation of the numbers entered*

*//Turning char's value to ints by subtracting 48*

*//reset count 2 to 0*

*// This is the validation of the numbers entered*

*//reset count 1 to 0*

*//Checking each guess against the random, if its correct, random value increases, count increases. Do 4 times*

*//If the count of match is equal to 4, 4 correct numbers in right position entered, won*

*//turned boolean b false, end game loop*

*//display congratulations message*

*//call function to increase wins*

*//if wrong value entered, display message*

*//if didnt win, display how many numbers entered are in the right position*

*//check every guess against all numbers, if there is a correct number in wrong position, increase count, change number value so it doesnt repeat*

*//do 4 times*

*//if didn't win display message, try again*

*//display how many correct numbers in wrong position*

*//Closing game loop, if count reaches 12 or b equals false*

*//Display sorry lost message*

*//call losses count function*

*//Give option to play again*

*//If player enters y, repeat loop to play again*

*//End option 2*

*//Beginning option 3*

*//calling level 3 function*

*//Option 4, display history of mastermind*

*//Open file where mastermind's history is*

*//Read data from file loop*

*//cout data*

*//CLose file*

*//end op 4*

*//op5 is to exit*

*//Display bye message*

*//call function to write wins and losses to file*

*//end op 5*

*//Default if a different or invalid value entered display invalid message*

*//Close switch*

*//Function for intro program*

*//Level 3 game function*

*//Declare Variables*

*//Calling function to generate random numbers*

*//Level 3 description*

*//Calling function to display pc's "player" name*

*//Setting game loop*

*//Closing number see function*

*//Resetting bool*

*//reset right answer counter*

*//reset right answer counter*

*//Counter for loop*

*//Copying random numbers to new loop so original is not lost*

*//storing the random numbers original in new loop*

*//CALLING FUNCTION TO INPUT GUESS*

*//For loop to fill 2 dimensional array*

*//Loop to check one by one correct guesses in the right place*

*//increase count if correct number in correct place*

*//if correct number in correct place change value of original*

*//Letting the user know how many numbers entered were correct in the right place*

*//Loop to check correct numbers in wrong place*

*//Check to see if player entered 4 correctly wins*

*//call win count function*

*//If player looses display message*

*//call lose count function*

*//Option to see the numbers*

*//If user presses y call see random numbers function*

*//CALLING FUNCTION TO SEE NUMBER*

*//Close game loop*

*//Call function that sorts guesses*

*//Function to see random numbers*

*//Loop to display random numbers, use while loop*

*//Random numbers generator function*

*//Declaring Variables*

*//Random numebers seed*

*//Setting loop for random numbers, since this is the hardest level 2 or more numbers will be equal*

*//Filling array with random numbers*

*//This makes sure there are at least 2 equal numbers because if there is no FALSE the loop wont stop!*

*//Close random numbers loop*

*//Input guess function level 3*

*//Having User input guess TURN UNTO A FUNCTION*

*//Declare Variables*

*//loop for input answer*

*//User inputs guess*

*//Check for string's length and valid inputs. If the input is less than 4 numbers have to input again*

*//This ensures that the data entered are digits and not letters*

*//A message of invalid input is displayed if the bool g1 becomes false due to an invalid input*

*//Turning string values to int*

*//Function to randomly select computer player*

*//Declare variables*

*//Randomly select computer player*

*//Function to calculate total wins*

*//Function to calculate total losses*

*//Function to calculate total wins and losses and output it to file*

*//Declare Variables*

*//Call win function*

*//Call loose function*

*//Calculating percent games won*

*//Open file to output data*

*//Outputting data to file*

*//Function to sort choices in third option of game*

*//Declare Variables*

*//Output not sorted array*

*//Sorting loops*

*//Output sorted array*

*//Open file to output data to*

*//Loop to copy data to file*

*//Closing File*

# 4 - Code

/\*

\* File: main.cpp

\* Author: Daniel

\*

\* Created on July 19, 2014, 1:15 PM

\*/

//System Library

#include <cstdlib>// Random numbers

#include <iostream>

#include <ctime>

#include <iomanip> //Output format Library

#include <string>//used for strings

#include <fstream>//to open file

using namespace std;

//User Libraries

//Global Constants

const int clP=1;

const int tdaG=5;

//Function Prototype

void intro(void);

void levU(string [][clP], string, string, int &, int &);

void seeRN(int []);

int radnG(int [], int);

int inpG(int [], int);

void compP(string [][clP]);

int wlC(string, int &);

int lC(string, int &);

void totwL(string, string, int &, int &);

int sortF(int [][tdaG], int);

//Execution Begins Here!

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

//Description of the program

intro();

cout<<endl;

//Declare Menu Variables

int const chom1=1,chom2=2,chom3=3,exitg=5,hisma=4;

int choiceg;

//Declaring variables for game part 1

char cgp1, cgp2, gmp1a; //Each one of this represents the numbers the player will enter, g1 for first number and so on...

int gp1, gp2;

int nump1o=0, nump2o=0; //Each one of this represents the random numbers that will be generated

int nump1, nump2;

//Declaring variables for game part 2

char cg1, cg2, cg3, cg4,gmppa; //Each one of this represents the numbers the player will enter, g1 for first number and so on...

int g1, g2, g3, g4;

int num1o, num2o, num3o, num4o; //Each one of this represents the random numbers that will be generated

int num1, num2, num3, num4;

//Declare Variables for open file, see history of mastermind

int opf1=127;

ifstream inputFile;

string name;

//Declare Variables for players name function

string playC[9][clP]={{"Mario from Honduras."},{"Mike from Germany."},{"Jose from Guatemala."},{"Jim from China."},{"Rachel from USA."},{"Lucia from Brazil."},

{"Ricardo from Mexico."},{"Tabang from Botswana."},{"Evelyn from South Africa."}};

//Declare Variables for w L functions

int w=0,l=0;

string naM1;

string naM;

//Counter common to both games

int match2=0,match=0, loopc=0; //c2=loop

//Random Numbers seed

srand(time(0));

cout<<"Please enter your name: ";

cin>>naM1;

//Loop to perform the game until user wants to exit

do

{

cout<<endl;

cout<<" 1. Easy Level (2 different numbers)"<<endl;

cout<<" 2. Hard level (4 different numbers)"<<endl;

cout<<" 3. Ultimate Level!!!!(2 or 3 numbers can be the same)"<<endl;

cout<<" 4. Mastermind's game history"<<endl;

cout<<" 5. Exit game"<<endl;

cin>>choiceg;

//Setting options using a switch

switch (choiceg)

{

case chom1:

{

do

{

do

{

//Setting the random numbers

nump1o=rand()%10;//This is random number 1

nump2o=rand()%9+1;//Random number 2

} while (nump1o==nump2o);

//Output the computer's numbers hidden

cout<<" The number is ? ? "<<endl;

cout<<""<<endl;

cout<<"You will play against ";

compP(playC);

cout<<endl;

cout<<"Start guessing now, remember that each number is between 0 and 10. Remember you have 7 opportunities"<<endl;

loopc=0;//restarting loop counter

bool c=true;//restating c's value

//Game loop

do

{

loopc++;// this is the count of the loop

nump1=nump1o;//taking the random number and making it equal to the variable I will use through the game

nump2=nump2o;

cout<<"Please enter your guess:"<<endl;

cin>>cgp1>>cgp2;

if ((cgp1>=48&&cgp1<=57)&&(cgp2>=48&&cgp2<=57))// This is the validation of the numbers entered, they must be numbers between 0 and 9

{

gp1=cgp1-48;//taking the char characters and turning them into numbers... use of ASCII code

gp2=cgp2-48;

match2=0; //reset count to 0

if ((gp1>=0&&gp1<10)&&(gp2>=0&&gp2<10)) // This is the validation of the numbers entered

{

match=0; //reset count to 0

//making sure if two numbers are equal in position and value a counter is started, if the counter hits 2, player wins

if (gp1==nump1)

{

nump1=11;

match++;

}

if (gp2==nump2){

nump2=12;

match++;

}

if (match == 2)

{

c=false;

cout<<"You won! Congratulations!"<<endl;

wlC(naM, w);

}

}

else

{

cout<<"One of the values you entered is invalid, please enter a different one."<<endl;

}

cout<<"You guessed: "<<match<<" numbers in the right position"<<endl;

if (gp1==nump2) //this is to verify each number against the other in case the player

{ //inputed the right number in the wrong position

nump2=30;

match2++;

}

if (gp2==nump1)

{

nump1=15;

match2++;

}

}

else

{

cout<<"One of the values you entered is invalid, please enter a different one."<<endl;

}

cout<<"You guessed: "<<match2<<" numbers in the wrong position"<<endl;

if ((c!=false)&&(gp1!=nump1||gp2!=nump2))//this makes sure that if the player didn't win a try again message is displayed

{

cout<<"Try again"<<endl;

}

} while ((loopc<7)&&(c!=false)); //while the count is not greater than 7 or c is false the loop will run

if (c!=false) //if c is true is because the player lost

{

cout<<"I am sorry, you lost."<<endl;

lC(naM1, l);

}

cout<<"Would you like to play again? enter Y for yes and N to go back to the main menu: "<<endl;

cin>>gmp1a;

} while (gmp1a=='y'||gmp1a=='Y');//if the player wants to play again y must be entered, everything else will be interpreted as if

//he doesn't want to play and the loop will end and he will be taken back to the menu

}break;

case chom2:

{

do

{

//Validation loop so that no 2 numbers are equal

do

{

//Setting the random numbers

num1o=rand()%10;//This is random number 1

num2o=rand()%10;//Random number 2

num3o=rand()%10;// Random number 3

num4o=rand()%10;// Random number 4

} while ((num1o==num2o||num1o==num3o||num1o==num4o)||(num2o==num3o||num2o==num4o)||(num3o==num4o));

cout<<" The number is ? ? ? ? "<<endl;

cout<<""<<endl;

cout<<"You will play against ";

compP(playC);

cout<<endl;

cout<<"Start guessing now, remember that each number is between 0 and 10. You will have 12 opportunities."<<endl;

loopc=0;//restarting loop count

bool b=true;//restating b's value as a bool

//Game loop

do

{

loopc++;// this is the count of the loop

num1=num1o;

num2=num2o;

num3=num3o;

num4=num4o;

cout<<"Please enter your guess:"<<endl;

cin>>cg1>>cg2>>cg3>>cg4;

if ((cg1>=48&&cg1<=57)&&(cg2>=48&&cg2<=57)&&(cg3>=48&&cg3<=57)&&(cg4>=48&&cg4<=57)) // This is the validation of the numbers entered

{

g1=cg1-48;

g2=cg2-48;

g3=cg3-48;

g4=cg4-48;

match2=0; //reset count to 0

if ((g1>=0&&g1<10)&&(g2>=0&&g2<10)&&(g3>=0&&g3<10)&&(g4>=0&&g4<10)) // This is the validation of the numbers entered

{

match=0; //reset count to 0

if (g1==num1)

{

num1=11;

match++;

}

if (g2==num2){

num2=12;

match++;

}

if (g3==num3)

{

num3=13;

match++;

}

if (g4==num4)

{

num4=14;

match++;

}

if (match == 4)

{

b=false;

cout<<"You won! Congratulations!"<<endl;

wlC(naM, w);

}

}

else

{

cout<<"One of the values you entered is invalid, please enter a different one."<<endl;

}

cout<<"You guessed: "<<match<<" numbers in the right position"<<endl;

if (g1==num2) //use else if cor each one

{

num2=30;

match2++;

} else if (g1==num3)

{

num3=31;

match2++;

} else if (g1==num4)

{

num4=32;

match2++;

}

if (g2==num1) //use else if cor each one

{

num1=15;

match2++;

} else if (g2==num3)

{

num3=16;

match2++;

} else if (g2==num4)

{

num4=17;

match2++;

}

if (g3==num1) //use else if cor each one

{

num1=18;

match2++;

} else if (g3==num2)

{

num2=19;

match2++;

} else if (g3==num4)

{

num4=20;

match2++;

}

if (g4==num1) //use else if cor each one

{

num1=21;

match2++;

} else if (g4==num2)

{

num2=22;

match2++;

} else if (g4==num3)

{

num3=23;

match2++;

}

else if ((b!=false)&&(g1!=num1||g2!=num2||g3!=num3||g4!=num4))

{

cout<<"Try again"<<endl;

}

}

else

{

cout<<"One of the values you entered is invalid, please enter a different one."<<endl;

}

cout<<"You guessed: "<<match2<<" numbers in the wrong position"<<endl;

} while ((loopc<12)&&(b!=false));

if (b!=false)

{

cout<<"I am sorry, you lost"<<endl;

lC(naM1, l);

}

cout<<"Would you like to play again? enter Y for yes, and N to return to the menu"<<endl;

cin>>gmppa;

} while (gmppa=='y'||gmppa=='Y');

} break;

case chom3:

{

levU(playC, naM, naM1, w, l);

} break;

case hisma:

{

inputFile.open("trial.txt");

for (int i=0;i<=opf1;i++)

{

inputFile>>name;

cout<<name<<" ";

}

cout<<endl;

cout<<endl;

inputFile.close();

}break;

case exitg:

{

cout<<"Good Bye!"<<endl;

totwL(naM, naM1, w, l);

}

break;

default:

cout<<"You have entered an invalid choice, choose again."<<endl;

}

} while (choiceg !=5);

return 0;

}

void intro (){

cout<<" MASTERMIND GAME "<<endl;

cout<<endl;

cout<<"This game is very simple. The program will generate a random number. Your job"<<endl;

cout<<"is to guess each one of the numbers that have been generated. You will be able"<<endl;

cout<<"to choose from 2 different levels, the easy one where you will guess 2 different numbers"<<endl;

cout<<"and the hard one where you will have to guess 4 different numbers. In the easy one you"<<endl;

cout<<"will have 7 opportunities to guess, in the hard one you will have 12. The game will"<<endl;

cout<<"give you feedback on how many numbers you guess correctly in the right spot, and how "<<endl;

cout<<"many numbers you guess correctly in the wrong spot. You will win the game if you guess "<<endl;

cout<<"all the numbers within the amount of opportunities provided, if not you will loose."<<endl;

cout<<"As a bonus has been included the history of Mastermind, if you wish to see it please"<<endl;

cout<<"select option 3. Have a great time!"<<endl;

cout<<"Now please choose which level you want"<<endl;

cout<<""<<endl;

}

void levU(string playC[][clP], string naM, string naM1, int &w, int &l){

//Declare Variables

const int arr2=4;

int rad2[arr2], check[arr2];

const int g2=4;

int arrg[g2]={0};

bool anseq=true, each;

int eq=0, cone=0, cone2=0, contg=0;

char seeN;

int guesS[10][tdaG];

//Calling function to generate random numbers

radnG(rad2, arr2);

cout<<setw(50)<<"The number is ? ? ? ?"<<endl;

cout<<""<<endl;

cout<<"You will play against ";

compP(playC);

cout<<endl;

cout<<"Start guessing now, remember that each number is between 0 and 10. This time you only have 10 opportunities."<<endl;

cout<<"In this level you will have one opportunity to see the random number if you wish."<<endl;

//Setting game loop

do

{

seeN='n';//Closing number see function

each=true;//Resetting bool

cone=0;//reset right answer counter

cone2=0;//reset right answer counter

contg++;//Counter for loop

//Copying random numbers to new loop so original is not lost

for (int i=0;i<4;i++)

{

check[i]=rad2[i];

}

//CALLING FUNCTION TO INPUT GUESS

inpG(arrg, g2);

//For loop to fill 2 dimensional array

guesS[contg-1][0]=contg;

for (int d=1;d<5;d++)

{

guesS[contg-1][d]=arrg[d-1];

}

//Loop to check one by one correct guesses in the right place

for (int i=0;i<4;i++)

{

if (arrg[i]==check[i])

{

cone++;

check[i]=12;

}

}

cout<<"You entered "<<cone<<" correct number in the right place."<<endl;//Letting the user know how many numbers entered were correct in the right place

//Loop to check correct numbers in wrong place

for (int i=0;i<4;i++)

{

for (int a=0;a<4;a++)

{

if (arrg[a]==check[i])

{

cone2++;

check[i]=11;

}

}

}

cout<<"You entered "<<cone2<<" correct numbers in the wrong place."<<endl;

if (cone==4)//Check to see if player entered 4 correctly wins

{

cout<<"You won! congratulations!"<<endl;

each=false;

wlC(naM, w);

}

if ((contg==10)&&(each==true))

{

cout<<"I am sorry, you lost."<<endl;

lC(naM1, l);

}

//Option to see the number

if ((contg==6) &&(each==true))

{

cout<<"If you want to see the number enter Y, else enter N: ";

cin>>seeN;

}

if (seeN=='Y'||seeN=='y')

{

//CALLING FUNCTION TO SEE NUMBER

seeRN(rad2);

cout<<endl;

}

} while((each==true)&&(contg<10));

sortF(guesS, contg);

}

void seeRN(int rad2[]){

//TURN THIS INTO A FUNCTION THAT DISPLAYS THE NUMBER IF PLAYER WANTS TO SEE IT

int i=0;

while (i<4)

{

cout<<rad2[i];

i++;

}

}

int radnG(int rad2[], int arr2){

//THIS WILL BE THE FUNCTION FOR THE HARDEST LEVEL

//Declaring Variables

int i=0,con;

bool rn;

srand(time(0));

//Setting loop for random numbers, since this is the hardest level 2 or more numbers will be equal

do{

for (int i=0;i<arr2;i++)

{

rn=true;

rad2[i]=rand()%10;//Filling array with random numbers

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

{

if (rad2[i]==rad2[r]) //This makes sure there are at least 2 equal numbers because if there is no FALSE the loop wont stop!

{

rn=false;

}

}

}

} while (rn==true);

// for (int i=0;i<4;i++)

// {

// cout<<rad2[i];

// }

// cout<<endl;

return rad2[arr2];

}

int inpG(int arrg[], int g2){

//Having User input guess TURN UNTO A FUNCTION

//Declare Variables

string guess;

bool g1;

//loop for input answer

do

{

g1=true;

cout<<"Please input your guess: ";

cin>>guess;

//Check for string's length and valid inputs. If the input is less than 4 numbers have to input again

if (guess.length()!=4)

{

g1=false;

}

for (int g=0;g<4;g++)

{

if (guess[g]<'0'||guess[g]>'9')//This ensures that the data entered are digits and not letters

{

g1=false;

}

}

if (g1==false)//A message of invalid input is displayed if the bool g1 becomes false due to an invalid input

{

cout<<"Your input is invalid, please enter a four digit guess."<<endl;

}

} while (g1==false);

//Turning string values to int

for (int i=0;i<4;i++)

{

arrg[i]=guess[i]-48;

}

return arrg[g2];

}

void compP(string playC[][clP]){

//Declare variables

int randN, i=15, count;

//Randomly select computer player

srand(time(0));

randN=rand()%9;

cout<<playC[randN][0];

}

int wlC(string naM, int &w){

cout<<"Good Job!"<<endl;

//Have user input name

return w++;

}

int lC(string naM1, int &l){

cout<<"Better luck next time."<<endl;

//Have user input name

return l++;

}

void totwL(string naM, string naM1, int &w, int &l){

//Declare Variables

int totw, totl;

float perW=0;

ofstream outputFile;

//Call win function

totw=wlC(naM, w);

//Call loose function

totl=lC(naM1,l);

//Calculating percent games won

perW=(totw/(totw+totl))\*100;

//Open file to output data

outputFile.open("Games won and Lost.txt");

//Outputting data to file

outputFile<<naM1<<" won "<<totw<<endl;

outputFile<<naM1<<" lost "<<totl<<endl;

outputFile<<naM1<<" won "<<perW<<"% of games."<<endl;

//Close file

outputFile.close();

}

int sortF(int guesS[][tdaG], int contg){

//Declare Variables

int temp;

ofstream outputFile;

bool swap;

//Output not sorted array

cout<<"Array not sorted."<<endl;

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

{

cout<<guesS[i][0]<<" ";

for (int d=1;d<5;d++)

{

cout<<guesS[i][d];

}

cout<<endl;

}

//Sorting loops

do

{

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

{

swap=false;

int q=1;

while (guesS[i][q]==guesS[i+1][q])

{

q++;

}

if(guesS[i][q]>guesS[i+1][q])

{

for (int g=0;g<5;g++)

{

temp = guesS[i][g];

guesS[i][g] = guesS[i + 1][g];

guesS[i + 1][g] = temp;

swap = true;

}

}

}

} while (swap);

//Output sorted array

cout<<"Sorted array."<<endl;

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

{

cout<<guesS[i][0]<<" ";

for (int d=1;d<5;d++)

{

cout<<guesS[i][d];

}

cout<<endl;

}

//Open file to output data to

outputFile.open("inputs sorted.txt");

//Loop to copy data to file

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

{

for (int d=0;d<5;d++)

{

outputFile<<guesS[i][d]<<" ";

}

outputFile<<endl;

}

//Closing File

outputFile.close();

}