Jose Morales

CSC-5 Intro C++

Dr. Lehr

07/21/2014

Project   
**MASTERMIND**

**Table of Contents**

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

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

Rules and Gameplay

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

Thoughts after writing the program

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

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

Initial approach ..........................................................................................................

Ideas implemented to meet challenges

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

Future improvements .............................................................................................................................. 5

**4 – Contents Covered**

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

**3- Flowchart** ............................................................................................................................... 7-15

**4- Code**

............................................................................................................................. 15-22

# 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. For my second project I am hoping to be able to work in a similar project but using more advanced C++ techniques such as arrays, more functions, and whatever else we learn in the remaining of the course.

# 2 - Development

## Strategy used to program the game:

### Initial approach:

At first I was not sure on how to program the game, I had ideas on how to get the user to input the two or four digits used to guess and validate them, but I was not sure if they would work. I started by trying to use the int value for the guesses, and having the player enter each number separated by blank space. This would allow me the possibility of evaluating each guess individually and making sure that they were numbers and within the range that was needed (0-9). I found this approach to be weary and unlikable for the user thus making him prone to disliking the game and not using it. Another challenge I found was validating each number not only against its individual position, but also against all the other positions without having the program crash. By crash I mean the following, let us assume the code consists of the numbers 2 and 4, and the user enters the numbers 4 4, the program would then compare the numbers entered and announce that two numbers had been entered correctly in the wrong position if it was not programmed correctly.

### Ideas implemented to meet challenges

I was able to fix the first problem by approaching the input values using the ASCII code values. I thank Brian for his help in this area since he was who thought of it first. Using this approach the user would enter his guess using char characters without the hassle of using spaces, and then I would to convert the char to int’s by subtracting forty-eight from each char (ASCII CODE). The second challenge was approached by using a counter for each number and by making sure that one of the numbers entered was correct but in the wrong position, its value would be changed during the rest of the process, being reseted to its original value by a loop at the beginning of the following “round” of guesses.

### Future improvements

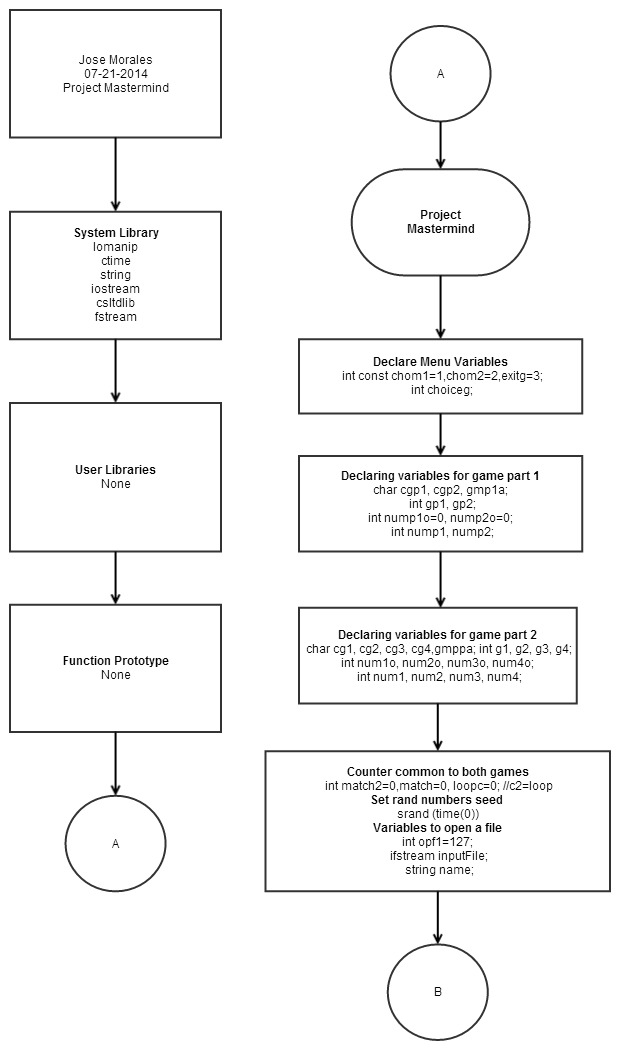
For the future I hope to be able to change the char characters input to array, I believe this will save me a lot of time and code space, it will also allow me to validate that the user only inputs four digits, not a necessary feature but just something I would like to add to the program. Also by using functions I will be able to reduce considerably the amount of code regarding if’s and the switch statements use.

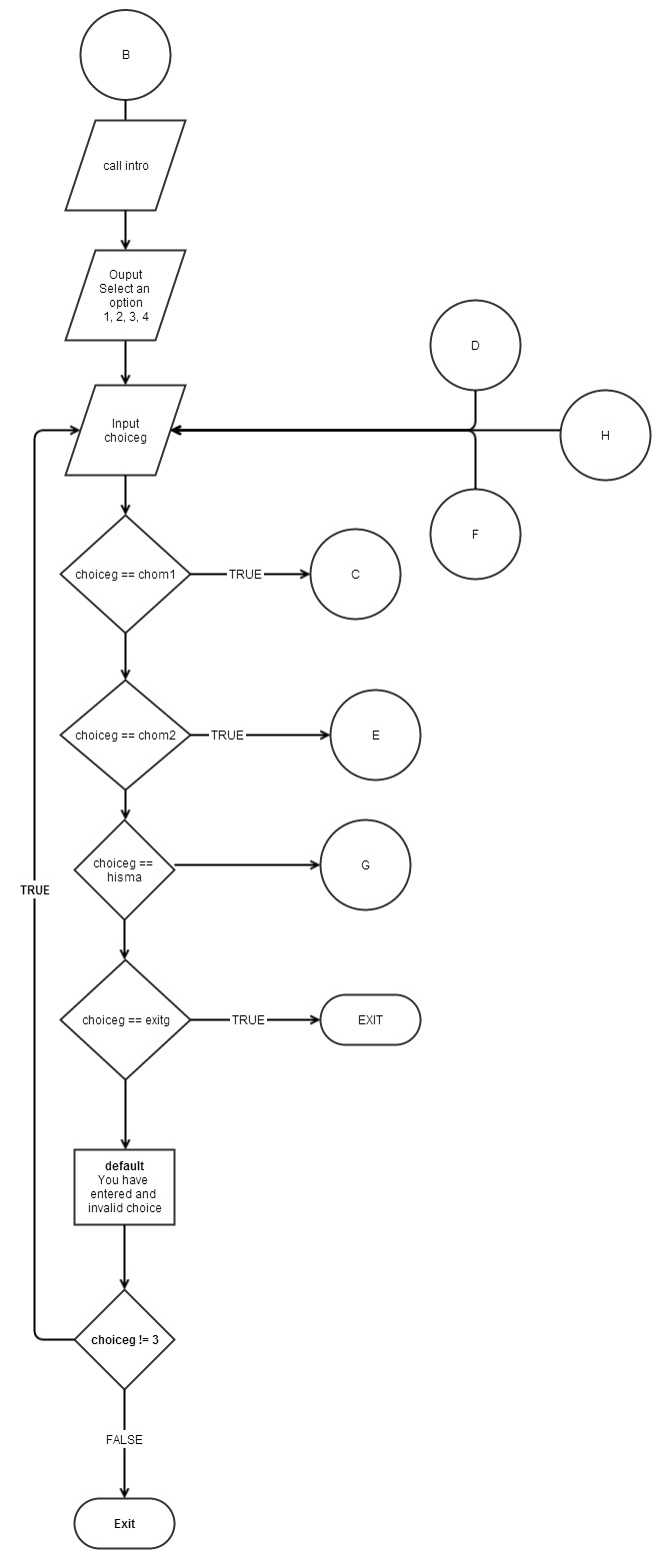
# CONTENTS COVERED

* Primitive data types:
  + Int = 34
  + Char = 33
  + Bool =195
  + String = 48
* System level variables
  + Cstdlib = 8
  + Iostream = 9
  + Ctime = 10
  + Iomanip = 11
  + String = 12
  + Fstream 13
* Operators
  + ++ = 95
  + == = 111
  + <= =208
  + && = 208
  + || = 309
  + >= 101
* Conditionals
  + If =101
  + If else = 131, 137
  + Do/while = 180,188
  + Validation =101
* Menu
  + Switch statement:= 71
* Functions
  + Void =369

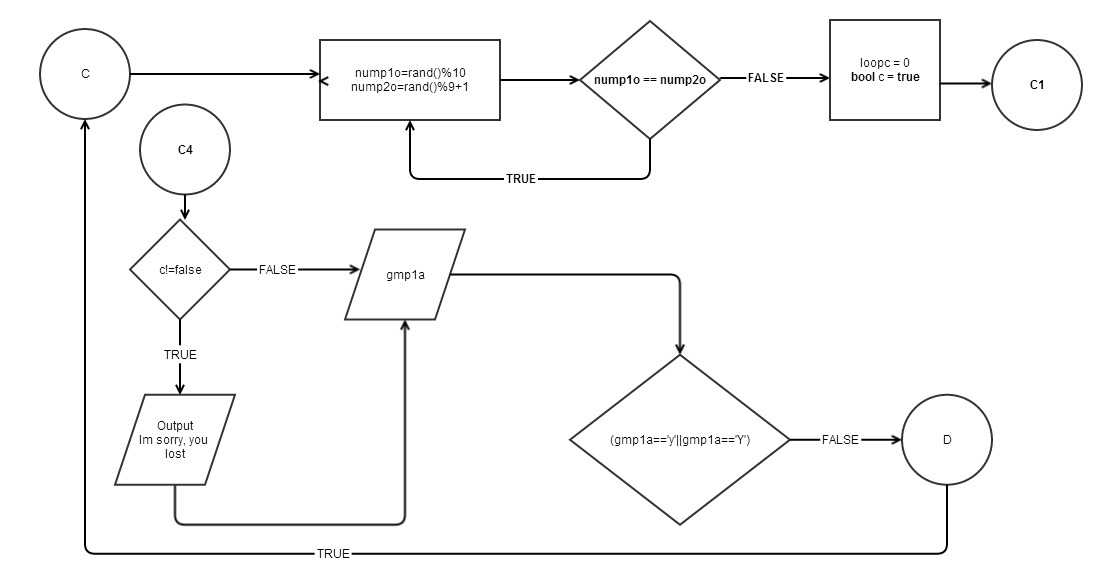
\* SIDE NOTE: The flowcharts will be uploaded to github for better visualization.

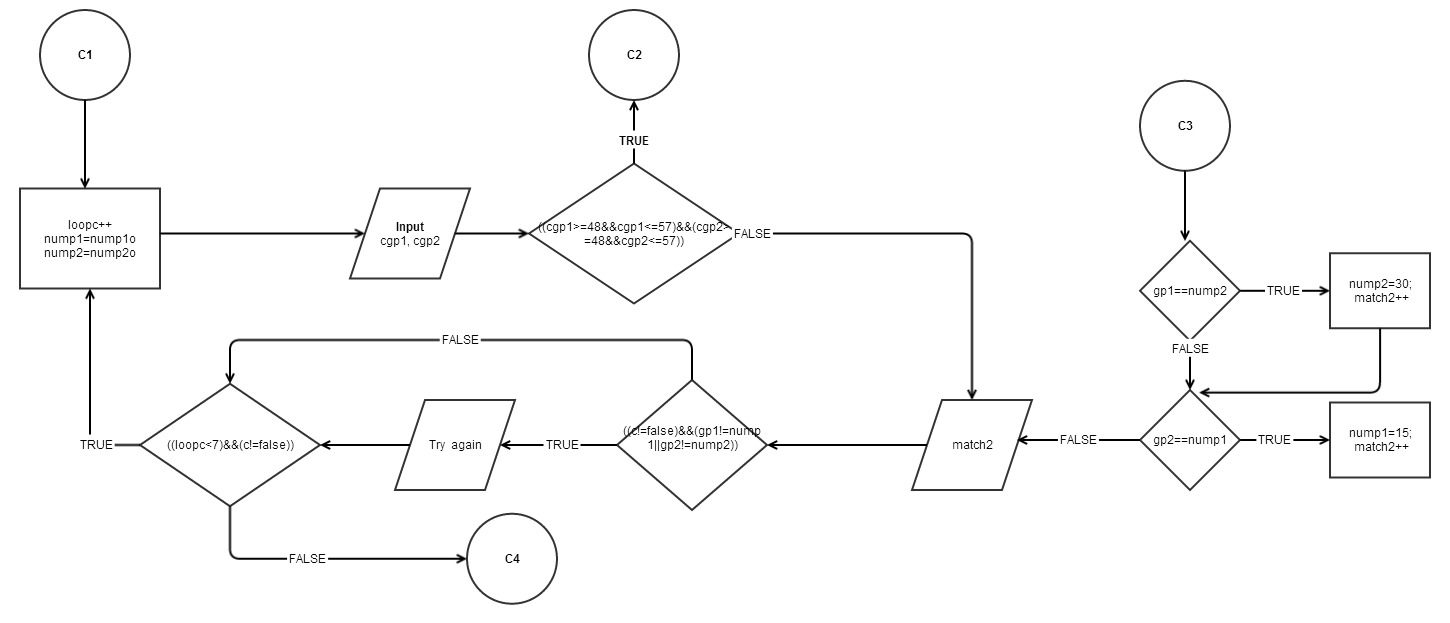
# FLOWCHARTS

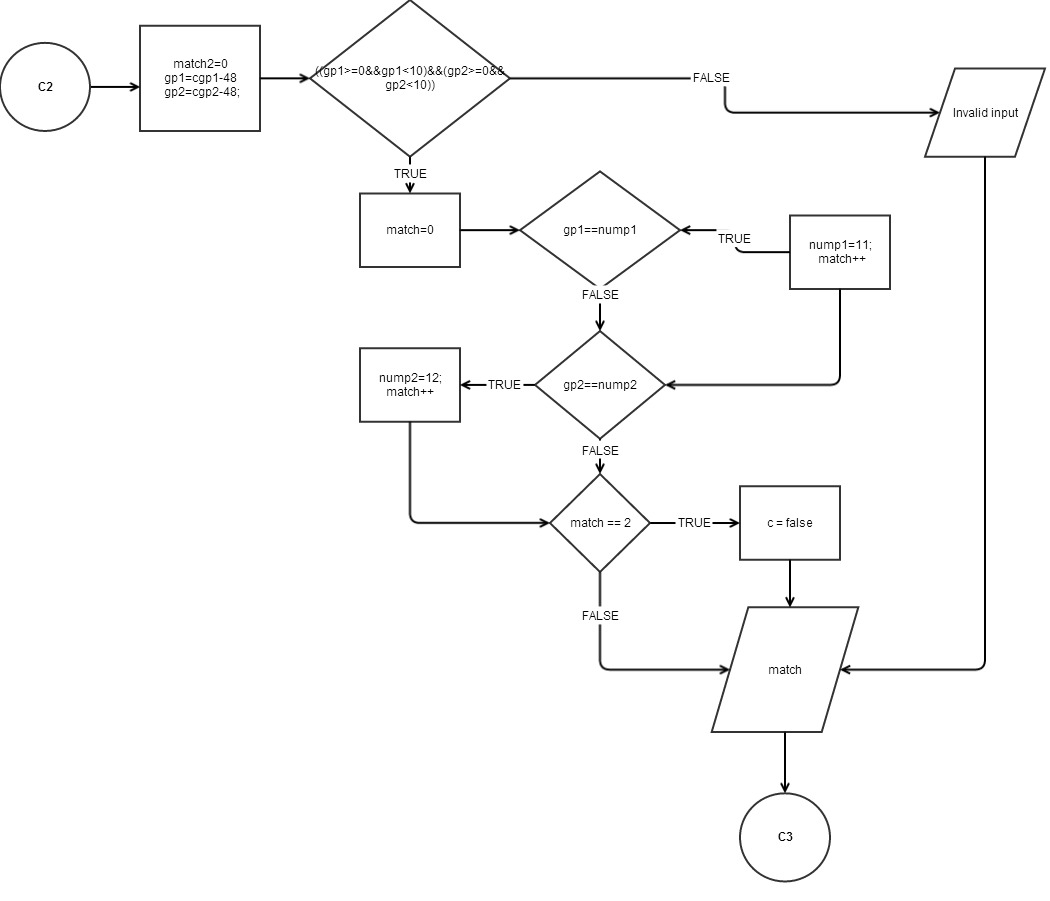




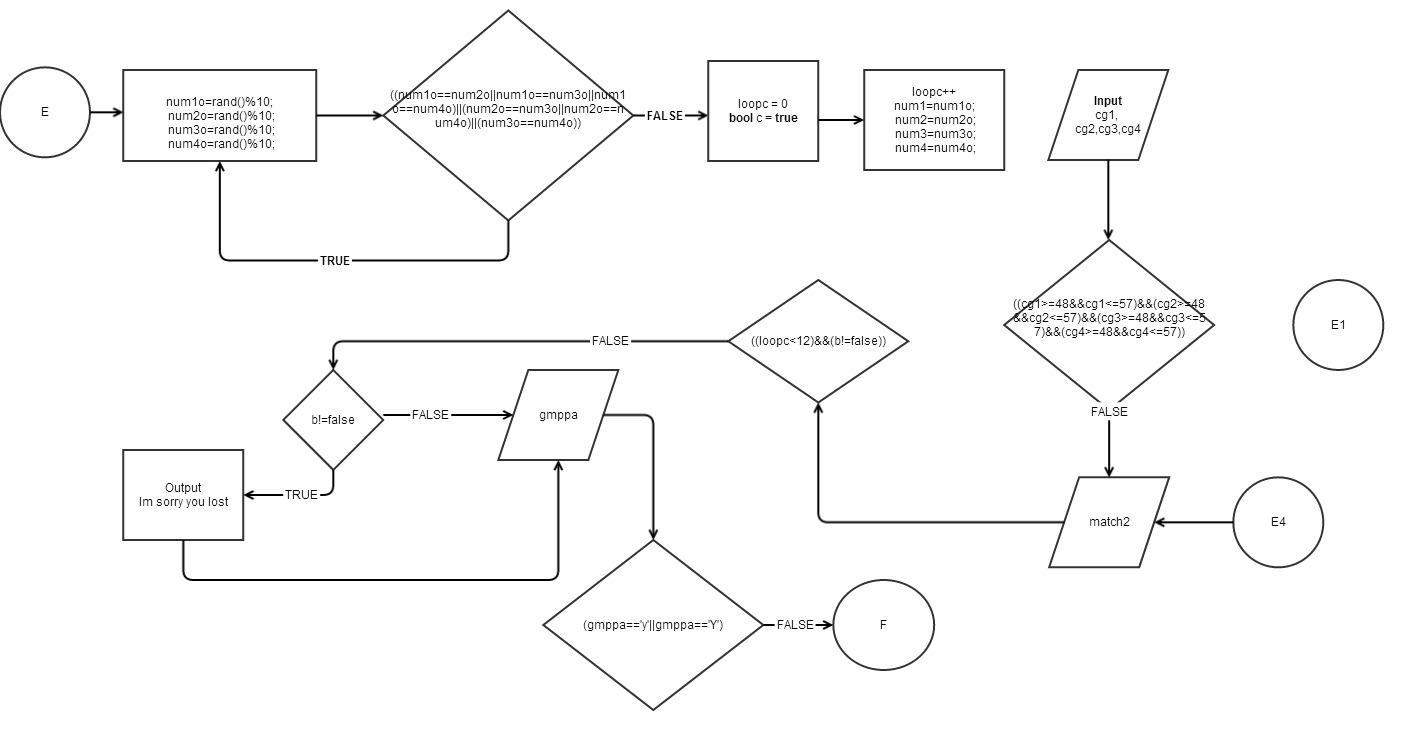
LEVEL 1

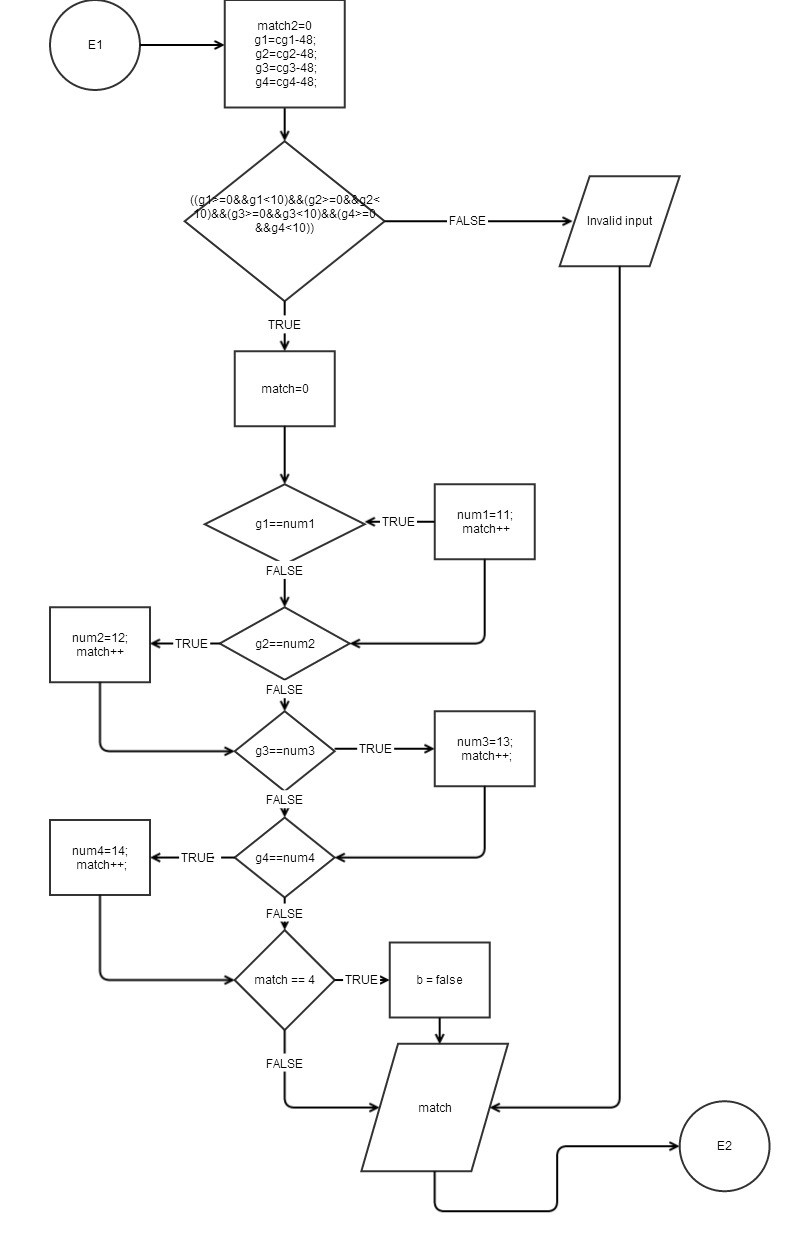


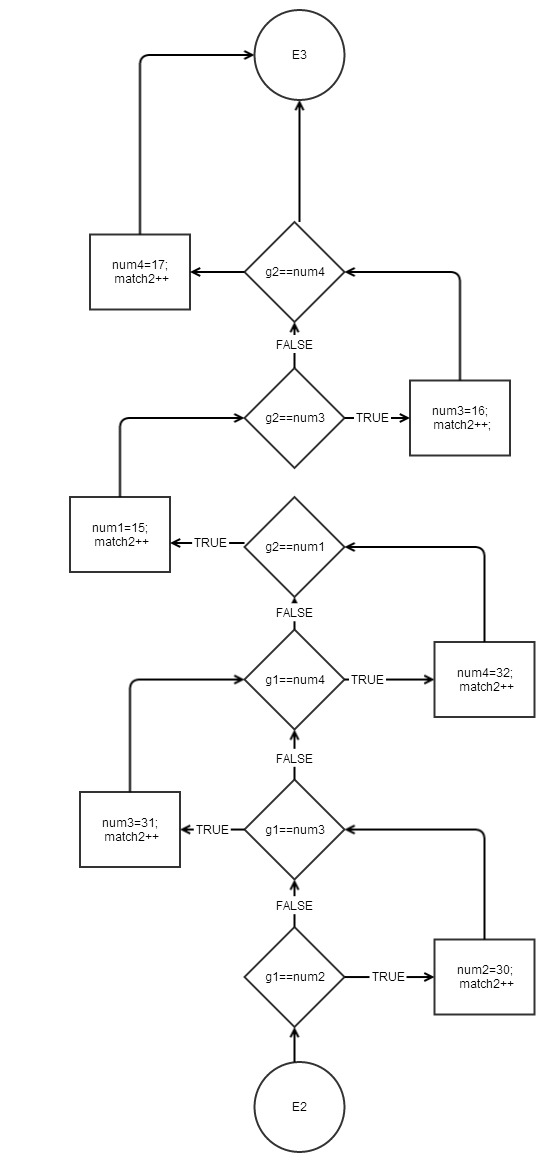


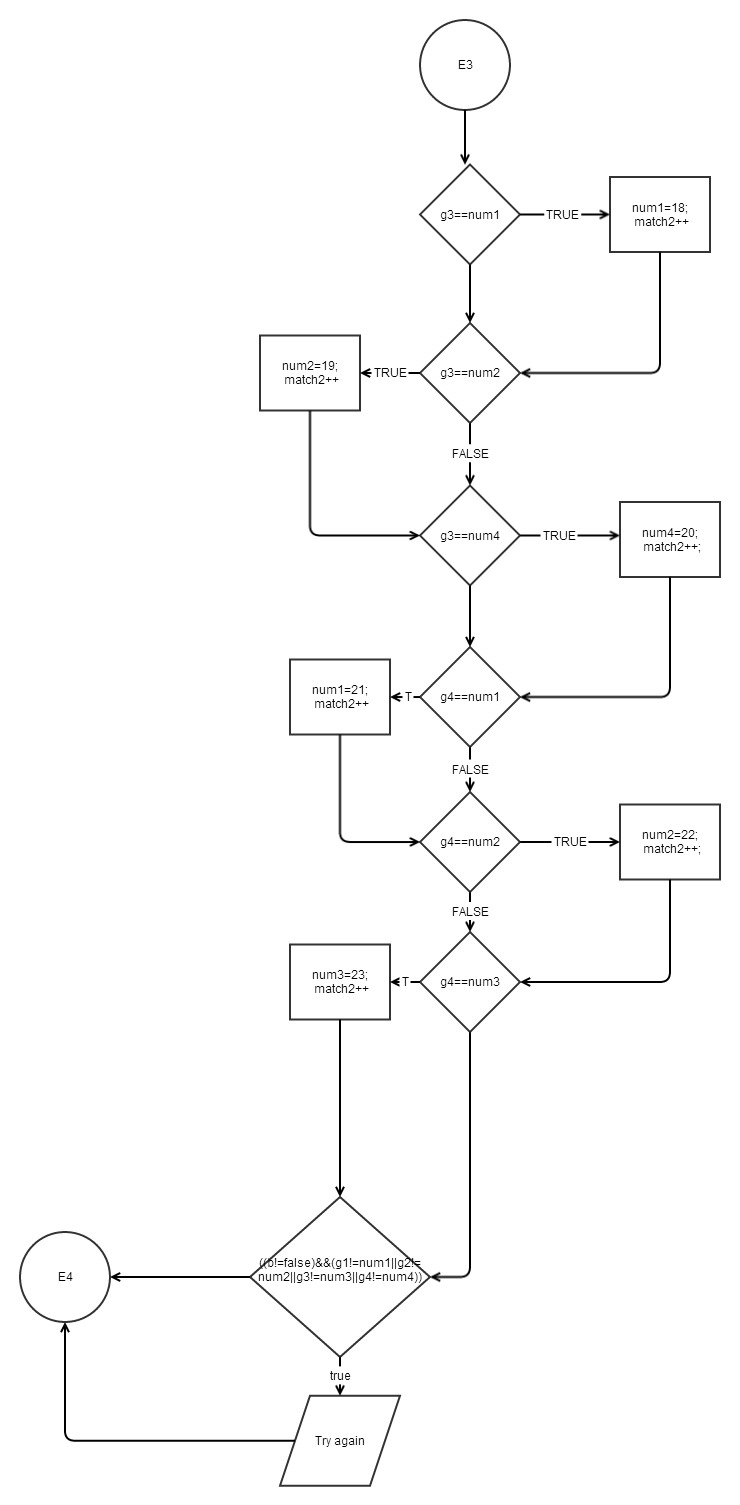


LEVEL 2









# 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

//Function Prototype

void intro(void);

//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,exitg=4,hisma=3;

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;

//Counter common to both games

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

//Random Numbers seed

srand(time(0));

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

do

{

cout<<" MASTERMIND "<<endl;

cout<<endl;

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

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

cout<<" 3. Mastermind's game history"<<endl;

cout<<" 4. 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<<"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;

}

}

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;

}

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<<"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;

}

}

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;

}

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 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;

}

break;

default:

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

}

} while (choiceg !=4);

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;

}