Lab Project 1

Mastermind

CSC-7-48309
Brandon Smith
10/17/2022

Introduction

Title: Mastermind

Mastermind is a game where the player must guess the sequence of colored pegs. The player can choose 4, 6, or 8 pegs and whether or not duplicate colors are allowed. After guessing a sequence of pegs, the user is told how many were in the correct color and the correct position and how many were the correct color but incorrect position.

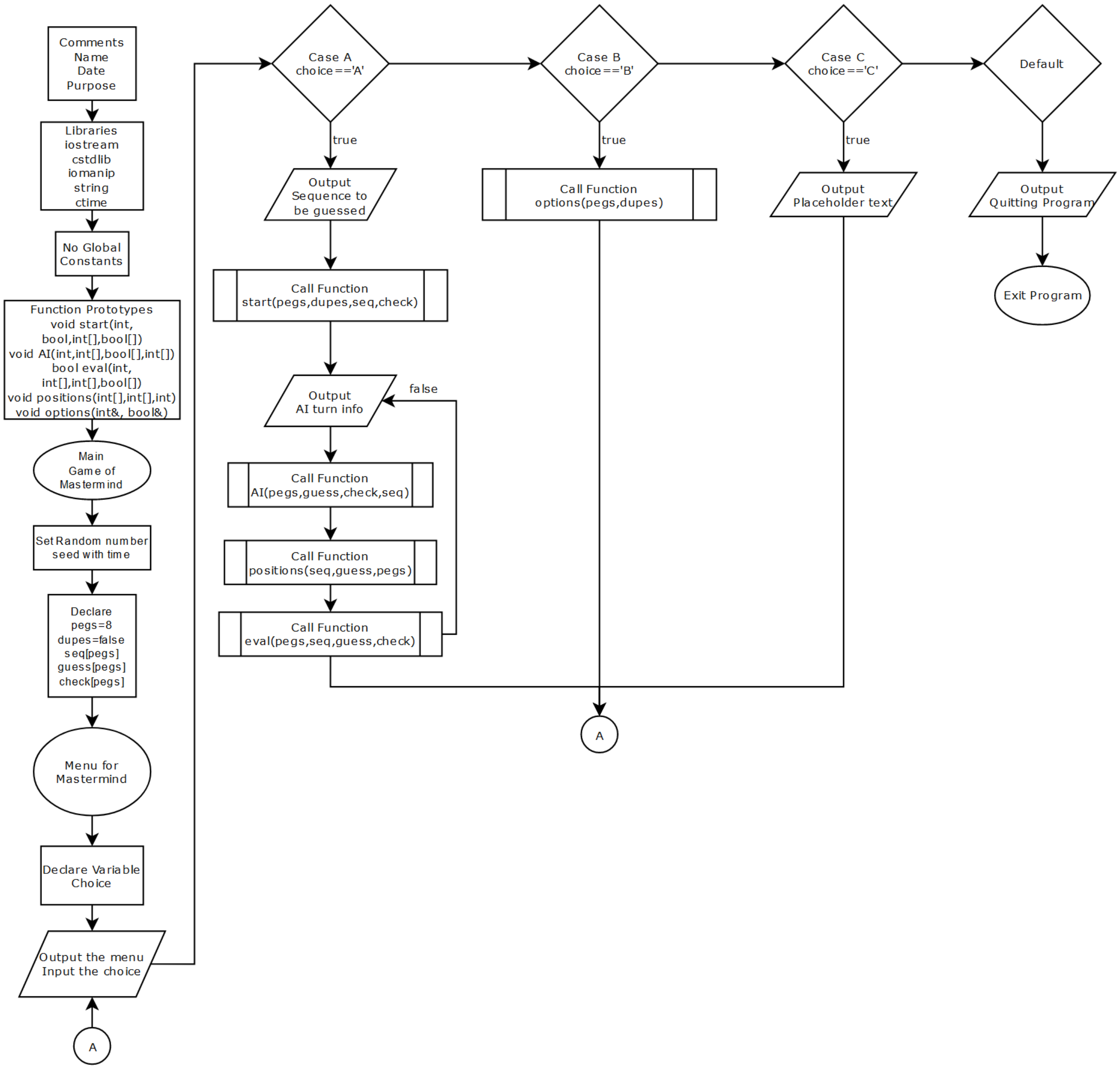
Summary

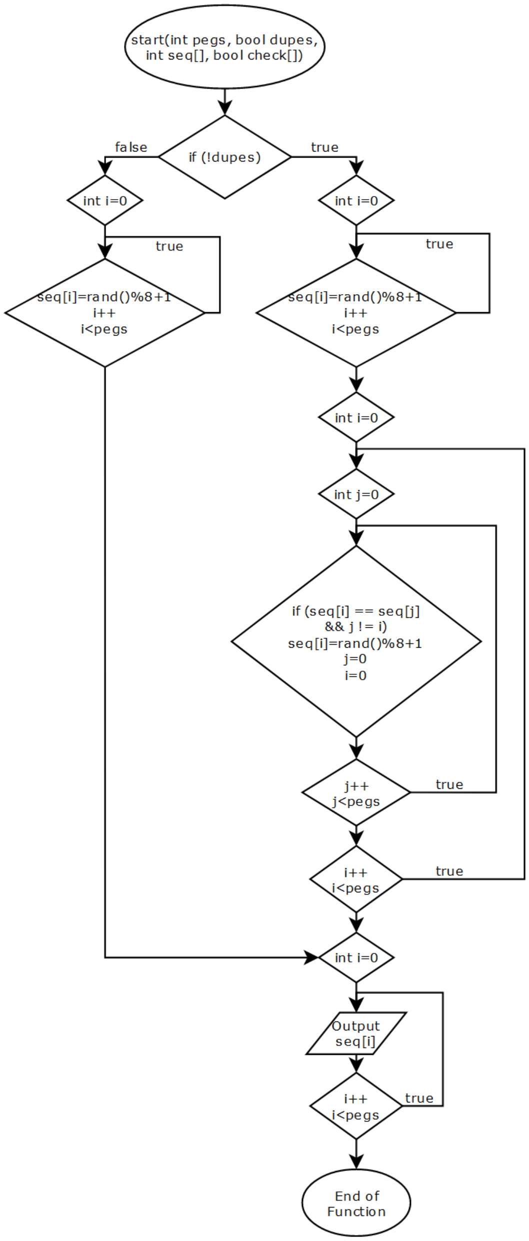
The program has 230 lines of code. This project had 3 separate versions. The final version for this project is still very basic and does not feature a fully functional AI. The main purpose of this first project was to build a working game, project 2 will feature an AI that uses algorithms to solve the game as efficiently as possible. There are still several bugs that need to be addressed in future versions for project 2, however this current version is able to set up and complete a full game of mastermind with several modifications for the user to change.

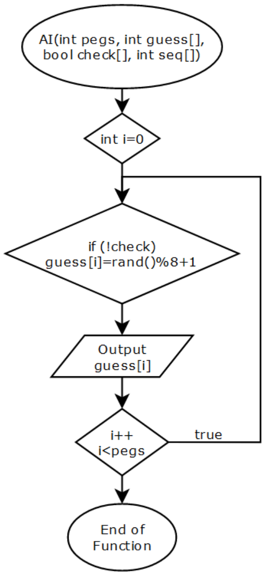
Description

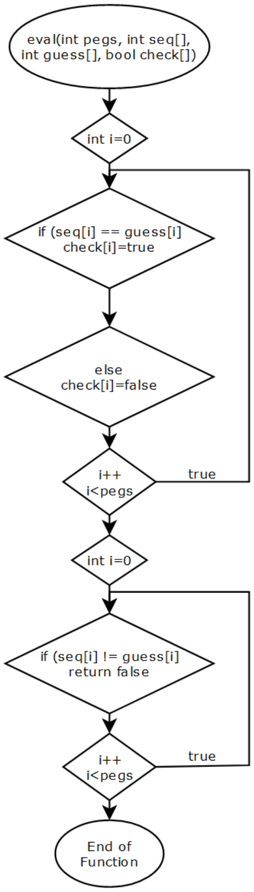
The primary focus of the program is to have an AI play a game of mastermind. The user is able to change the number of pegs and allow or disallow duplicates.

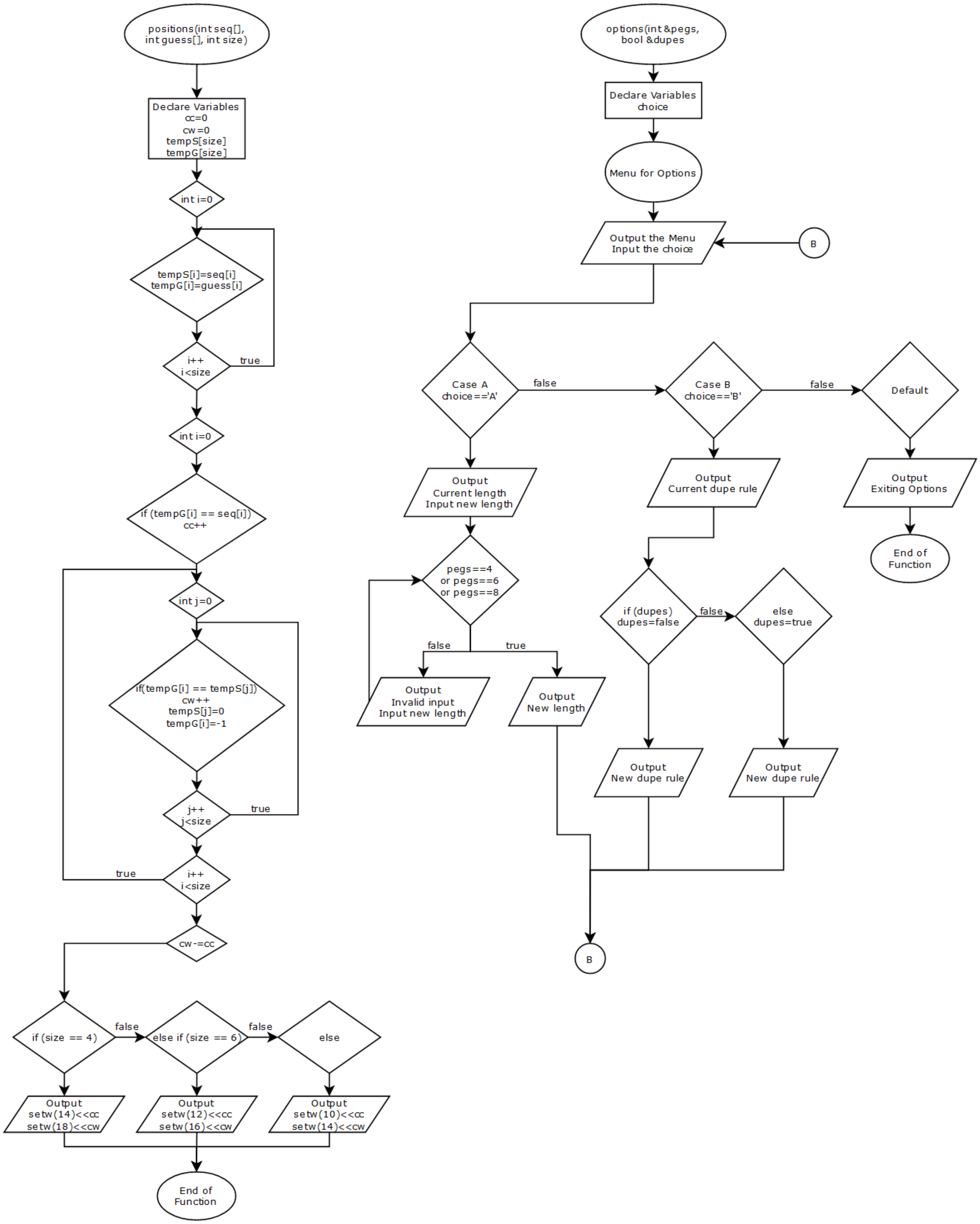
Flowchart


Comments
Name
Date
Purpose
Libraries
iostream
cstdlib
iomanip
string
ctime
No Global
Constants
Function Prototypes
void start(int,
bool,int[],bool[])
void AI(int,int[],bool[],int[])
bool eval(int,
int[],int[],bool[])
void positions(int[],int[],int)
void options(int&, bool&)
Main
Game of
Mastermind
Set Random number
seed with time
Declare
pegs=8
dupes=false
seq[pegs]
guess[pegs]
check[pegs]
Menu for
Mastermind
Declare Variable
Choice
Output the menu
Input the choice
A
Case A
choice=='A'
true
Output
Sequence to
be guessed
Call Function
start(pegs,dupes,seq,check)
Output
AI turn info
false
Call Function
AI(pegs,guess,check,seq)
Call Function
positions(seq,guess,pegs)
Call Function
eval(pegs,seq,guess,check)
Case B
choice=='B'
true
Call Function
options(pegs,dupes)
A
Case C
choice=='C'
true
Output
Placeholder text
Default
Output
Quitting Program
Exit Program


start(int pegs, bool dupes, int seq[], bool check[])
if (!dupes)
false
true
int i=0
int i=0
true
true
seq[i]=rand()%8+1
i++
i<pegs
seq[i]=rand()%8+1
i++
i<pegs
int i=0
int j=0
if (seq[i] == seq[j]
&& j != i)
seq[i]=rand()%8+1
j=0
i=0
j++
j<pegs
true
i++
i<pegs
true
int i=0
Output
seq[i]
i++
i<pegs
true
End of
Function


AI(int pegs, int guess[], bool check[], int seq[])
int i=0
if (!check)
guess[i]=rand()%8+1
Output
guess[i]
i++
i<pegs
true
End of
Function


eval(int pegs, int seq[], int guess[], bool check[])
int i=0
if (seq[i] == guess[i]
check[i]=true
else
check[i]=false
i++
i<pegs
true
int i=0
if (seq[i] != guess[i]
return false
i++
i<pegs
true
End of
Function

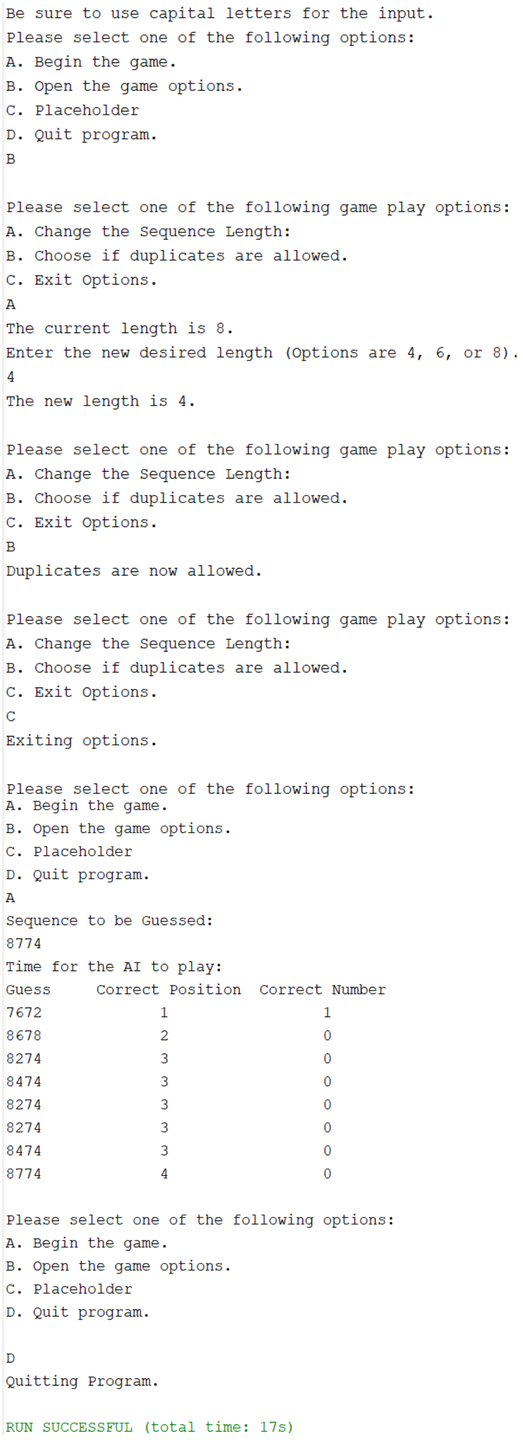

positions(int seq[], int guess[], int size)
Declare Variables
cc=0
cw=0
tempS[size]
tempG[size]
int i=0
tempS[i]=seq[i]
tempG[i]=guess[i]
i++
i<size
true
int i=0
if (tempG[i] == seq[i])
cc++
int j=0
if(tempG[i] == tempS[j])
cw++
tempS[j]=0
tempG[i]=-1
j++
j<size
true
i++
i<size
true
cw-=cc
if (size == 4)
false
else if (size == 6)
false
else
Output
setw(14)<<cc
setw(18)<<cw
Output
setw(12)<<cc
setw(16)<<cw
Output
setw(10)<<cc
setw(14)<<cw
End of Function
options(int &pegs, bool &dupes
Declare Variables
choice
Menu for Options
Output the Menu
Input the choice
B
Case A
choice=='A'
false
Case B
choice=='B'
false
Default
Output
Current length
Input new length
Output
Current dupe rule
Output
Exiting Options
End of Function
pegs==4
or pegs==6
or pegs==8
false
true
Output
Invalid input
Input new length
Output
New length
if (dupes)
dupes=false
false
else
dupes=true
Output
New dupe rule
Output
New dupe rule
B

Pseudo Code

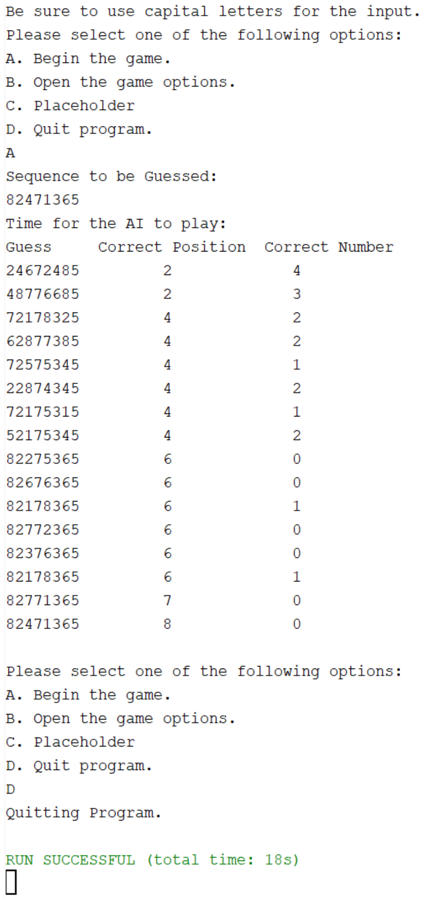
```
Initialize

Do
Display the game menu
    If A is entered
        Output
        Call start game function
        Output
        Do
            Call AI function
            Call positions function
        While eval function is false
        Break
    If B is entered
        Call options function
        Break
    If C is entered
        Output (this is a placeholder case)
        Break
    If any other input is entered
        Output
        Break
While choice is A, B, or C

Return 0

Functions

Void start
    If no dupes are allowed
        Set sequence using for loop and srand
        Check for dupes using nested for loops and if statements
    Else (dupes are allowed)
        Set sequence using for loop and srand
    Output sequence using for loop
        Set bool check array to false

Void AI
    Use a for loop to step through each element of the arrays
        If current check bool element is false
            Set new srand value for current guess element
        Output guess for current element
```

```
Bool eval
    Use a for loop to step through each element of the guess and sequence arrays
        If current sequence element is equal to current guess element
            Current check bool element is set to true
        Else
            Current check bool is set to false
    Use a for loop to step through each element of the guess and sequence arrays
        If current sequence element is not equal to current guess element
            Return false

Void positions
    Initialize variables
    Copy sequence and guess arrays
    Check and record correct positions/numbers using nested for loops and if statements
    Calculate correct position but wrong number
    Output

Void options
    Initialize variables
    Do
    Output options menu
        If A is entered
            Output current length
            Cin new sequence length
            Validate sequence length
            Output new length
            Break
        If B is entered
            If dupes are allowed
                Bool dupes=false
                Output new rule
            If dupes are not allowed
                Bool dupes=true
                Output new rule
            Break
        If any other input is entered
            Output
            Break
    While choice is A or B
```

Proof of Working Code



```
Be sure to use capital letters for the input.
Please select one of the following options:
A. Begin the game.
B. Open the game options.
C. Placeholder
D. Quit program.
B

Please select one of the following game play options:
A. Change the Sequence Length:
B. Choose if duplicates are allowed.
C. Exit Options.
A
The current length is 8.
Enter the new desired length (Options are 4, 6, or 8).
4
The new length is 4.

Please select one of the following game play options:
A. Change the Sequence Length:
B. Choose if duplicates are allowed.
C. Exit Options.
B
Duplicates are now allowed.

Please select one of the following game play options:
A. Change the Sequence Length:
B. Choose if duplicates are allowed.
C. Exit Options.
C
Exiting options.

Please select one of the following options:
A. Begin the game.
B. Open the game options.
C. Placeholder
D. Quit program.
A
Sequence to be Guessed:
8774
Time for the AI to play:
Guess     Correct Position  Correct Number
7672            1                  1
8678            2                  0
8274            3                  0
8474            3                  0
8274            3                  0
8274            3                  0
8474            3                  0
8774            4                  0

Please select one of the following options:
A. Begin the game.
B. Open the game options.
C. Placeholder
D. Quit program.

D
Quitting Program.

RUN SUCCESSFUL (total time: 17s)
```



```
Be sure to use capital letters for the input.
Please select one of the following options:
A. Begin the game.
B. Open the game options.
C. Placeholder
D. Quit program.
A
Sequence to be Guessed:
82471365
Time for the AI to play:
Guess     Correct Position  Correct Number
24672485        2              4
48776685        2              3
72178325        4              2
62877385        4              2
72575345        4              1
22874345        4              2
72175315        4              1
52175345        4              2
82275365        6              0
82676365        6              0
82178365        6              1
82772365        6              0
82376365        6              0
82178365        6              1
82771365        7              0
82471365        8              0

Please select one of the following options:
A. Begin the game.
B. Open the game options.
C. Placeholder
D. Quit program.
D
Quitting Program.

RUN SUCCESSFUL (total time: 18s)
```

Program

```
//System Libraries
#include <iostream>
#include <cstdlib>
#include <iomanip>
#include <string>
#include <ctime>
using namespace std;

//Function Prototypes
void start(int, bool,int[],bool[]);   //Function for game setup
void AI(int,int[],bool[],int[]);                //Function for AI's turn
bool eval(int, int[],int[],bool[]);   //Function to evaluate AI's guess
void positions(int[],int[],int);     //Function to calculate and output positions
void options(int&, bool&);                 //Function to change options

//Execution Begins Here
int main(int argc, char** argv) {
        //Initialize the Random Number Seed
        srand(static_cast<unsigned int>(time(0)));

        //Declare Variables
        int pegs=8;      //Default pegs
        bool dupes=false;   //Default dupes
        int seq[pegs];  //Sequence array (move to start function)
        int guess[pegs];          //AI guess array (move to start function)
        bool check[pegs];   //Bool array to check each digit (move to start function)

        //Game Menu
        char choice;
        cout<<"Be sure to use capital letters for the input.";  //Automatically capitalize input for
user?
        do{
        cout<<endl<<"Please select one of the following options:"<<endl;
        cout<<"A. Begin the game."<<endl;
        cout<<"B. Open the game options."<<endl;
        cout<<"C. Placeholder"<<endl;
        cout<<"D. Quit program."<<endl;
        cin>>choice;
        switch(choice){
        case 'A':
                //Set up the game
                //Move all cout statements to respective functions for v4
                cout<<"Sequence to be Guessed:"<<endl;
                start(pegs,dupes,seq,check);
                //AI's turn
                //The current "AI" is fully random, but in Project 2
                //the AI will be completely rewritten and use strategy to guess
                cout<<"Time for the AI to play:"<<endl;
                cout<<"Guess            Correct Position  Correct Number"<<endl;
                do{
                AI(pegs,guess,check,seq);

                //Display the current game stats
                positions(seq,guess,pegs);

                }while(eval(pegs,seq,guess,check)==false);
                break;

        case 'B':
                options(pegs,dupes);
                break;

        case 'C':
                cout<<"This is a placeholder (maybe display stats?)"<<endl;
                break;

        default:
                cout<<"Quitting Program."<<endl;
                break;
        }
        }while (choice == 'A' || choice == 'B' || choice == 'C');

        //Exit program
        return 0;
}

void start(int pegs,bool dupes,int seq[],bool check[]){
        //Without Dupes (make more efficient)
        if (!dupes){
        for (int i=0; i<pegs; i++){
        seq[i]=rand()%8+1;
        }

        for (int i=0; i<pegs; i++){        //Possibly make its own function
        for (int j=0; j<pegs; j++){
                if (seq[i] == seq[j] && j != i){
                seq[i]=rand()%8+1;
```

```
                j=0;
                i=0;
                }
        }
        }
        }

        //With Dupes
        else{
        for (int i=0; i<pegs; i++){
        seq[i]=rand()%8+1;
        }
        }

        //Display Sequence
        for (int i=0; i<pegs; i++){
        cout<<seq[i];

        //Temporary line to set the AI's check to false by default
        //Will remove when implementing a full AI
        check[i]=false;
        }

        cout<<endl;
}

void AI(int pegs,int guess[],bool check[],int seq[]){
        for (int i=0; i<pegs; i++){
        if (!check[i]){
        guess[i]=rand()%8+1;
        }
        cout<<guess[i];
        }
}

bool eval(int pegs,int seq[],int guess[],bool check[]){
        for (int i=0; i<pegs; i++){
        if (seq[i] == guess[i]){
        check[i]=true;
        }else{   //Not needed?
        check[i]=false;
        }
        }
        for (int i=0; i<pegs; i++){
```

```
        if (seq[i]!=guess[i])return false;
        //Add else return true
        }
}

void positions (int seq[], int guess[], int size){
        //Note this entire function will be changed in Project 2 with the proper AI
        int cc=0;       //Correct number and position
        int cw=0;       //Correct number wrong position
        int tempS[size];//Temp seq array copy
        int tempG[size];//Temp guess array copy

        //Copy arrays
        for (int i=0; i<size; i++){
        tempS[i]=seq[i];
        tempG[i]=guess[i];
        }

        //Check correct positions and numbers
        for (int i=0; i<size; i++){
        if (tempG[i] == seq[i])cc++;

        for (int j=0; j<size; j++){
        if (tempG[i] == tempS[j]){
                cw++;
                tempS[j]=0;
                tempG[i]=-1;
        }
        }
        }
        cw-=cc;

        //Output results
        if (size == 4){
        cout<<setw(14)<<cc;
        cout<<setw(18)<<cw<<endl;
        }else if (size == 6){
        cout<<setw(12)<<cc;
        cout<<setw(16)<<cw<<endl;
        }else{
        cout<<setw(10)<<cc;
        cout<<setw(14)<<cw<<endl;
        }
}
```

```
void options (int &pegs,bool &dupes){
    char choice;
    do{
    cout<<endl<<"Please select one of the following game play options:"<<endl;
    cout<<"A. Change the Sequence Length:"<<endl;    //Add current option to output
    cout<<"B. Choose if duplicates are allowed."<<endl; //Add current option to output
    cout<<"C. Exit Options."<<endl;
    cin>>choice;
    switch(choice){
    case 'A':
        //Get new length and validate input
        cout<<"The current length is "<<pegs<<"."<<endl;
        cout<<"Enter the new desired length (Options are 4, 6, or 8)."<<endl;
        cin>>pegs;
        while (pegs != 4 && pegs != 6 && pegs != 8){
        cout<<"Invalid length, please select 4, 6, or 8."<<endl;
        cin>>pegs;
        }
        cout<<"The new length is "<<pegs<<"."<<endl;
        break;

    case 'B':
        //Swap dupes bool and output new rule
        if (dupes){
        dupes=false;
        cout<<"Duplicates are no longer allowed."<<endl;
        }else{
        dupes=true;
        cout<<"Duplicates are now allowed."<<endl;
        }
        break;

    default:
        cout<<"Exiting options."<<endl;
        break;
    }
    }while (choice == 'A' || choice == 'B');
}
```