**Take Home Program 2– Due on or before Friday 9/21/2018**

1. **Lottery.cpp –** Objective: Parallel Array

Write a program to randomly generates a lottery of a five-digit number ranging from 0 to 9 and store in W**inningDigits** array. The program asks the user to enter a five-digit number in a range from 0-9 and store in P**layerDigits** array.

Example of matching digits -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Computer | 7 | 4 | 9 | 1 | 3 |
| User | 1 | 2 | 9 | 5 | 3 |

**Sample output 1:**

Enter the 5 digits of your lottery number, separated by blanks: 9 3 4 4 1

Winning number: 30766

Your number : 93441

You have 0 matching digit(s).

**Sample output 2:**

Enter the 5 digits of your lottery number, separated by blanks: 5 5 4 1 7

Winning number: 46419

Your number : 55417

You have 2 matching digit(s).

**// Function prototypes**

**bool PlayerDigits(int[]);**

Called by main; to reads in the digits of the player and stores in an array that is passed to it.

If all digits are valid, return true. If not, false is returned.

**void WinningDigits(int[]);**

Called by main; to randomly generate the digits of the winning lottery number and stores in an array that is passed to it.

**int countMatches(int[], int[]);**

Called by main; counts the number of digits that matched in the user array and random array passed to it.

**string getNumberString(int[]);**

Returns a string made up of the digits in the integer array that is passed to it.

**Copy and paste your program (source) code and the outputs after this line**

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

/\*

Lottery.cpp

Inola Cohen

CSIT 839 - 26953

Purpose: to write a program that randomly

generates a lottery of a five-digit number

ranging from 0-9 and store in array. The

program asks the user to enter a fve-digit

number in a range from 0-9 and store in another array

\*/

/\* Function Prototypes \*/

void WinningDigits(int[], int);

bool PlayerDigits(int[], int);

int countMatches(int[], int[], int);

void getNumberString(int);

//#include “stdafx.h” // doesn’t work on mac

#include <iostream>

#include <string>

#include <ctime>

using namespace std;

int main()

{

int PLAYER\_SIZE = 5, WINNING\_SIZE = 5, i;

int winningDigits[WINNING\_SIZE], playerDigits[PLAYER\_SIZE];

int amountMatching;

string answer;

do

{

cout << "Enter the 5 digits of your lottery number (between 0-9), seperated by blanks: ";

PlayerDigits(playerDigits, 5);

WinningDigits(winningDigits, 5);

countMatches(winningDigits, playerDigits, 5);

cout << "Winning number: ";

for (i = 0; i < WINNING\_SIZE; i++)

{

winningDigits[i] = rand() % 10;

cout << winningDigits[i] << " ";

}

cout << endl;

cout << "Your number: ";

for (i = 0; i < PLAYER\_SIZE; i++)

{

cout << playerDigits[i] << " ";

}

cout << endl << endl;

amountMatching = countMatches(winningDigits, playerDigits, 5);

getNumberString(amountMatching);

cout << "\nDo you want to try again? (Y/N) ";

cin >> answer;

cout << endl;

} while (answer == "Y" || answer == "y" || answer == "Yes" || answer == "yes" || answer == "YES");

return 0;

}

void WinningDigits(int winningNumbers[], int WINNING\_SIZE)

{

unsigned int seed;

seed = time(0);

srand(seed);

}

bool PlayerDigits(int playerNumbers[], int PLAYER\_SIZE)

{

int i;

for ( i = 0; i < PLAYER\_SIZE; i++)

{

cin >> playerNumbers[i];

}

for (i = 0; i < PLAYER\_SIZE; i++)

{

//cout << playerNumbers[i] << " "; //for validation

while (playerNumbers[i] < 0 || playerNumbers[i] > 9)

{

cout << "Please re-enter a valid number for entry #"

<< i + 1 << ": ";

cin >> playerNumbers[i];

}

}

return true;

}

int countMatches(int winningNumbers[], int playerNumbers[], int ARRAY\_SIZE)

{

int i, count = 0;

for (i = 0; i < ARRAY\_SIZE; i ++)

{

if (winningNumbers[i] == playerNumbers[i])

{

count++;

}

}

return count;

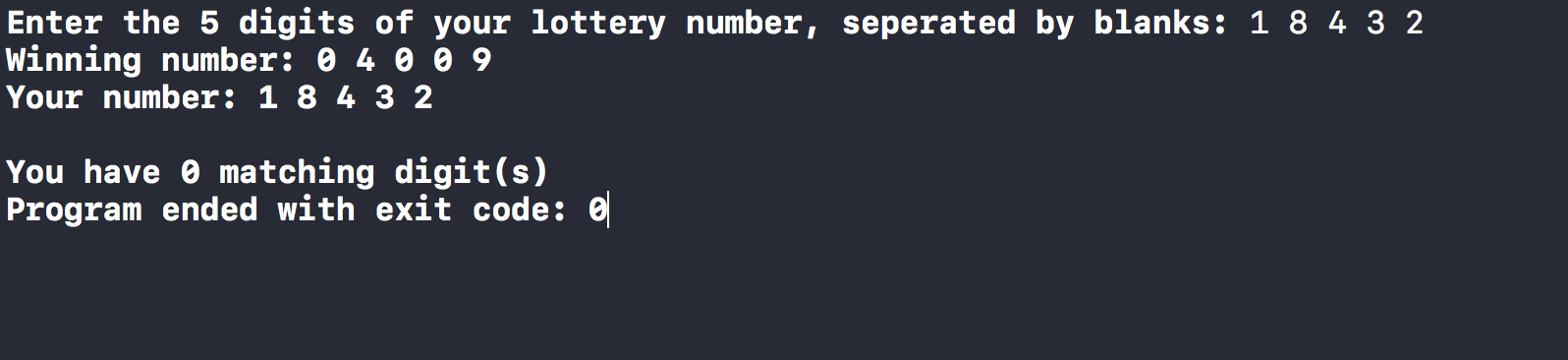
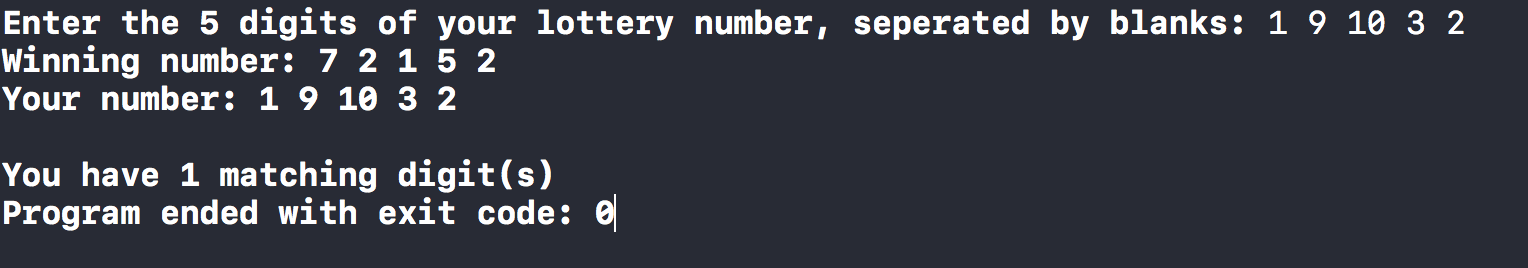
}

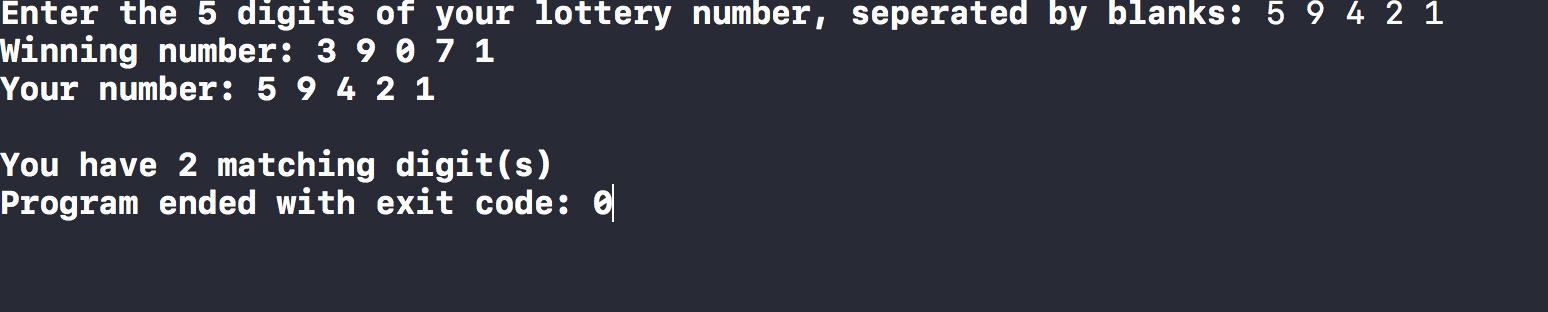
void getNumberString(int AmountMatching)

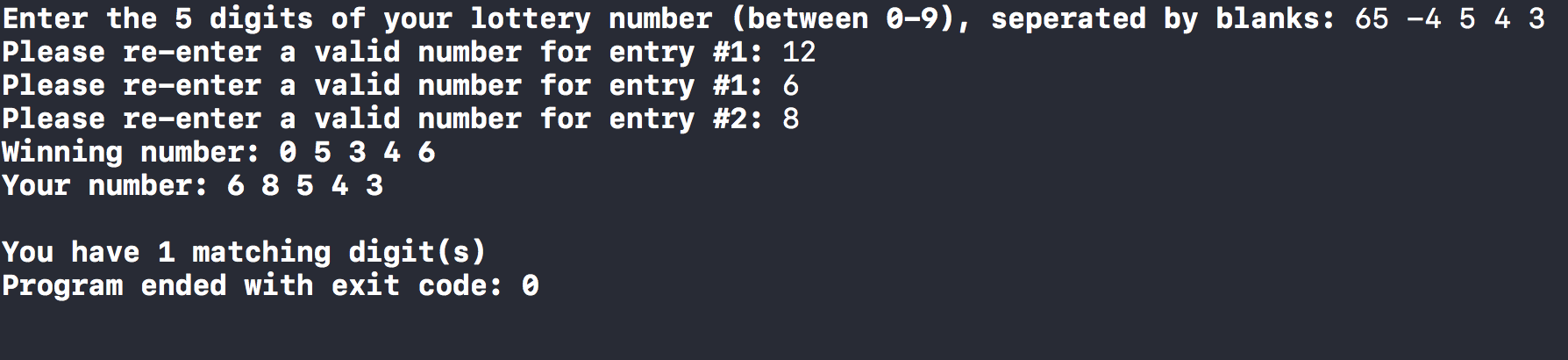
{

cout << "You have " << AmountMatching << " matching digit(s)" << endl;

}

****

****

****