**Programming Project Report**

Name: Riley Nadwodny

Date: 10/28/19

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

The primary goal of this programming assignment was to give us experience with defining functions and calling them with different parameter set-ups. We were given a partial implementation of a “Horoscope Generator” program, and from there, we had to define the rest of the functions that were needed in the program. It would ask the user for personal info such as birth month, day, and year, calculate the user’s karma, and generate a random horoscope to be output to the user based on the information given by the user.

For the birth date, the user is entered into a “while” loop that checks that the user’s input is in the valid range, which is from 1900 to 2019. For the birth month, a “while” loop occurs yet again, this time checking that the user’s input is a valid month selection. If the input is invalid, they have to continue to enter in inputs until the input any month between Jan – Dec. Lastly, the same kind of “while” loop happens for the birth day, but this time, the valid range is from 0 to 31; if the input is outside of that range, they have to continue to input values until their input is within the valid range.

**Design:**

For the design of this program, half of the program was supplied by the professor, John Gauch. The program consists of 10 functions, 11 including “int main()”, and a handful of data types for data inputs and outputs. 6 of the functions operate in order to make different predictions for the user’s horoscope, using a variety of “switch” statements and “if else-if” statements. There are a few different algorithms used throughout the program; one such algorithm is “int choice = (karma \* 17) % 5;”. Choice is assigned either a 0, 1, 2, 3, or 4, and is followed by a switch statement with cases from 0-4. Depending on the remainder that “choice” is set to will determine which case in the switch statement occurs.

In one particular function I created in this program, I had a little bit of fun with the decision making; the algorithm in the “Family” function is “int choice = ((year \* karma) \* rand()) / 100000;”. This algorithm multiplies the birth year and randomly generated karma together, multiplies the result by a random number, and then divides it by 100,000. I found that while doing testing for this, I was getting some very large results when trying to use 1,000 and 10,000 as the divider. As a result, I ultimately went with 100,000 because that brought the results down to a more manageable number. The rest of the function is a series of “if, else-if” statements with the bounds for the statements being 0-1000, 1000-2000, 2000-3000, 3000-4000, 4000-5000, and 5000+.

Splitting each prediction into a separate function made it very easy to write a function, test it, do debugging, and then move on to the next function. I definitely preferred it for this project over trying to write 350+ lines of code in “main”; for larger programs, I can see the value of breaking parts of the program into different functions for easier testing and debugging, as well as readability of the program as a whole.

**Implementation**

To start, half of the sample code was provided by the professor, John Gauch. In this sample code, three of the prediction functions were already coded, the karma generation, the “while” loops for the user inputs, and sections of the code were commented out for future function implementation and variable declaration, as well as different areas to call functions. The task assigned was to fill out the rest of the program with the remaining three functions including declaration, parameter declaration, filling out the body of the remaining functions, and returning outputs. After sitting down to flesh out the remainder of the program, overall time to completion was about 6-8 hours including testing a variety of inputs and debugging functions, as well as debugging the overall program to make sure everything was working as intended.

**Testing**

For testing the program, various birth dates along with a number of people spoken hello to on the given day, have been used as inputs for testing.

Sample input & output #1: My birthday

Enter the year you were born [1900..2019]: 1998

Enter the month you were born [Jan..Dec]: Oct

Enter the day you were born [1..31]: 08

How many people have you said hello to today? 3

Your astrological sign is Libra

I think it is singles night at the local pub.

You can save more money by buying less beer.

It looks like a chance for some overtime.

You and your family have some bright times ahead.

You may want to see a doctor...

Login to every website you can find today, and you will find out the true meaning of the internet and life today. (Hint: The answer may be 42.)

Sample input & output #2: Random birthday

Enter the year you were born [1900..2019]: 2002

Enter the month you were born [Jan..Dec]: May

Enter the day you were born [1..31]: 13

How many people have you said hello to today? 42

Your astrological sign is Taurus

Maybe your date would like to see a romantic comedy?

This is a good time to your parents for a bigger allowance.

Your boss looks generous today, time to ask for a raise.

Family prediction is impossible.

Beware of an impending sickness; perhaps start taking allergy and flu medication.

Great things are ahead! You are going to inherit a TRILLION dollars, become president of the universe, and recreate dinosaurs and create a real Jurassic Park!

Sample input & output #3: A 2nd random birthday

Enter the year you were born [1900..2019]: 200

Enter the year you were born [1900..2019]: 300

Enter the year you were born [1900..2019]: 2000

Enter the month you were born [Jan..Dec]: October

Enter the month you were born [Jan..Dec]: December

Enter the month you were born [Jan..Dec]: Nov

Enter the day you were born [1..31]: 43

Enter the day you were born [1..31]: 0

Enter the day you were born [1..31]: 32

Enter the day you were born [1..31]: 4

How many people have you said hello to today? 7

Your astrological sign is Scorpio

You should sit by the fire and read a good romance novel.

This is a good time to your parents for a bigger allowance.

Your boss looks generous today, time to ask for a raise.

There's some tension within your family; talk to them about what may be going on.

You are in an incredibly well period of health!

Great things are ahead! You are going to inherit a TRILLION dollars, become president of the universe, and recreate dinosaurs and create a real Jurassic Park!

One problem with the program that is a known issue is that when anything other than a number is input for the year of the user’s birth, the program enters into an infinite loop, as shown below.

Enter the year you were born [1900..2019]: Oct

Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]: Enter the year you were born [1900..2019]:

The same occurs on when the input for the user’s birth day is being checked. The while loop is thrown into an infinite loop, resulting to the user having to use ^C to stop the program in both cases.

Enter the year you were born [1900..2019]: 1998

Enter the month you were born [Jan..Dec]: 200

Enter the month you were born [Jan..Dec]: 300

Enter the month you were born [Jan..Dec]: 400

Enter the month you were born [Jan..Dec]: Oct

Enter the day you were born [1..31]: Month

Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]: Enter the day you were born [1..31]:

**Conclusions**

Overall, this project has been a success. In doing this project, I have learned much more about valid function declarations and parameter declarations, as well as function calls, const parameters in function declarations, and reference parameters. I probably would not do anything different next time, as there isn’t much to change. One thing I would likely do is spend time figuring out how to prevent the program from going into an infinite loop in the two cases shown in the testing section of this report. However, since that wasn’t an objective or worry for this project, I decided to not worry about it too much this time around. The time to complete the project, including this lab report, took a total of about 10-12 hours cumulatively after all said and done.