Name: _________________________Session: _______
Programming II
Lab Exercise 5/31/2023

1. The Powerball lottery is a numbers game. The first part requires drawing 5 numbers that range from 1 to 69 from a drum of white balls. The second part involves drawing a red ball (the Powerball) from another drum that has the numbers 1 to 26. This forms the complete Powerball result. For example, your result may be 12, 21, 34, 39, and 62 with the Powerball being 21. Write a simulation that first generates a Powerball result (your pick). The simulation then starts generating Powerball results until a result matching the original result (your pick) is obtained. Keep track of the number of results it takes for this to occur. Assuming there are 2 Powerball drawing each week, your simulation should report the time in years that it takes for you to win the Powerball lottery.

2. Suppose you translate a number it into a string of digits spelled out, one word for each digit, followed by a single space. For example, the number 407 becomes the digit string: "FOUR ZERO SEVEN". Write a program to accept a positive number and print out its DIGIT STRING. The program should work for decimal numbers as shown in the Sample Run.

```
Sample Run
Enter a positive number: 407.8
The Digit String = FOUR ZERO SEVEN . EIGHT
```

3. While numbers do not feel emotion, people do respond emotionally when numbers behave consistently in surprising and amusing ways. Happy Numbers can be reduced to 1 with a simple formula.

   To determine if a number is happy, or not, follow these simple rules with any number:
   - If the number is a single digit, square it.
   - If the result or the original number has multiple digits, take each digit by itself and square the digit and add the squares of the digits.
   - Repeat until you get to the number 1 or you find the results are repeating.

   Happy numbers, and other numeric patterns, can be identified programmatically with code. The first step to creating pseudo code to evaluate happy numbers is to remind yourself how to determine if a number is happy or not. You take any number, break the number into single digits, then square each digit and add up the sum of all you multiplications. You repeat this process until either you get the number 1 (the number you started with is happy) or your results match a pattern that indicates an unhappy number (the sequence of results starts repeating).

   Your task is to write a program that will find all of the happy numbers from 1 to 1000. I recommend writing a function isHappy that is passed an integer and returns True if the integer is happy and False if the number is sad.

4. In number theory, a perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. For instance, 6 has divisors 1, 2 and 3 (excluding itself), and 1 + 2 + 3 = 6, so 6 is a perfect number. Write a program that will report all of the perfect numbers less than or equal to 10000. Do you see any pattern in the perfect numbers?