**Name: Session:**

**Programming I**

**Lab Exercise 10.28.2020**

For each of these problems, you are to submit your documented source code.

1. Consider the following program which contains 4 functions:

def f(x) :

return g(x) + math.sqrt(h(x))

def g(x) :

return 4 \* h(x)

def h(x) :

return x \* x + k(x) – 1

def k(x) :

return 2 \* (x + 1)

print f(2)

What is printed to the screen? \_\_\_\_\_\_\_\_\_\_\_

1. Write a function

def isLeapYear(year)

that tests whether a year is a leap year: that is, a year with 366 days. The rules for a leap are as follows:

* A year is a leap year if it is evenly divisible by 4 and
* It is not divisible by 100
* Unless it is divisible by 400.

Use the following code to test your function:

print isLeapYear(1996) #True

print isLeapYear(1900) #False

print isLeapYear(2000) #True

print isLeapYear(1993) #False

1. Write a program that converts a Roman number such as MCMLXXVIII to its decimal number representation. *Hint:* First write a function or create a dictionary that yields the numeric value of each of the letters. Then use the following algorithm (in pseudocode):

total = 0

While the roman number string is not empty

If value(first character) is at least value(second character), or the string has length 1

Add value(first character) to total.

Remove the character.

Else

Add the difference, value(second character) - value(first character), to total.

Remove both characters.

1. 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?

Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What are the perfect numbers <= 10000?

Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. In mathematics, a **Mersenne prime** is a prime number that is one less than a power of two. That is, it is a prime number of the form *Mn* = 2*n* − 1 for some integer *n*. They are named after Marin Mersenne, a French Minim friar, who studied them in the early 17th century. Specifically, A Mersenne prime is a Mersenne number, i.e., a number of the form

|  |
| --- |
| M_n=2^n-1, |

that is prime. In order for M_n to be prime, n must itself be prime. It should be noted that there are only 51 known Mersenne primes. The last 17 were found by GIMPS (Great Internet Mersenne Primes Search) using distributed computing techniques. The 8th one was found by Leonard Euler in 1772. The 51st one was found in 2018 with a value of 282589933 – 1 (24862048 digits).

Write a program that will report the first 8 Mersenne primes.

Answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_