Matthew Abruzzese Ott

111520701

Python Project 1 Report

AMS 595

Part 1

This program has you input a number n, and using if, elif, and else statements, it outputs "negative one" if n = -1, "positive one" if n = 1, "zero" if n = 0, and "other value" if n = any other value.

Part 2

This program uses a while loop to generate a list of the Fibonacci sequence for all values that do not exceed 4,000,000, starting at 1. While the sum of the previous two terms is less than 4,000,000, that sum becomes the next term in the Fibonacci sequence and is appended to the list. The while loop iterates until the newest term exceeds 4,000,000.

Part 3

This program uses list comprehension to take a value x*y, for all x in 100-999 and y in 100-999 to see which products are palindromes using a string condition. This list is then sorted, and the last value in the sorted list is printed, i.e., the largest palindrome.

Part 4

A)

This program determines whether a number is prime or not using a for loop within if, else statements. For n greater than one, a number i is generated from a range and if n is divisible by i, then n is not prime. Otherwise, the number n is prime.

B)

This program finds all the factors of a number n using a for loop. It iterates through all numbers between 1 and n, and if n is divisible by i, then i is a factor of n, and i is appended to the list of factors for n.

C)

This program finds the prime factorization of a number n using a while loop. Starting at i = 2, while i*i <= n, if n % i, then i is not a prime factor. If n //= i, then i is a prime factor of n, and i is appended to the list of prime factors. After the loop terminates, if n > 1, append n to the list of prime factors.

Part 5

This program uses iteration (for loop and nested for loop) to generate the nth row of Pascal's triangle. Input the row n, and for i in the range 1 through n, start the row with 1 always, and for j in the range 1 through i, if i = n, append it to the nth row list. Then save the current value of the nth row in the nth row list and reiterate until complete. Then return the list which yields the nth row of Pascal's triangle.