Homework 1

Michael Brodskiy
Professor: M. Fanaei

January 24, 2023

Listing 1: Problem 1

```
"""
 * ============================================================================
 *
 *        Filename:  HW1Prob1Brodskiy.py
 *        Assignment: Homework 1 Problem 1
 *        Title: MPG Tracker
 *
 *     Description:  Takes in miles driven and gallons used until a
      negative number is entered, then outputs an mpg value
 *
 *         Version:  1.0
 *         Created:  01/24/2023
 *        Revision:  N/A
 *          Python:  Python 3.9.2
 *
 *          Author:  M. Brodskiy
 *
 * ============================================================================
"""

galUsed = 0
milDriven = 0
sumMiles = 0
sumGal = 0

while 1 != 2:
    galUsed = float(input("Enter the number of gallons used (any
       negative number to end): "))

    if galUsed < 0:
        break

    milDriven = float(input("Enter the number of miles driven: "))
    print(f"The miles/gallon for this tank was {milDriven /
       galUsed:.6f}.")
    sumMiles += milDriven
    sumGal += galUsed

print(f"The overall average miles/gallon for the above tankfuls
   was {sumMiles / sumGal:6f}.")
```

Listing 2: Problem 2

```
"""
 * ============================================================================
 *
 *        Filename:  HW1Prob2Brodskiy.py
 *        Assignment: Homework 1 Problem 2
 *        Title: Palindrome Verification
 *
 *     Description:  Takes in a number and determines whether it is
    a palindrome
 *
 *         Version:  1.0
 *         Created:  01/24/2023
 *        Revision:  N/A
 *          Python:  Python 3.9.2
 *
 *          Author:  M. Brodskiy
 *
 * ============================================================================
"""

num = int(input("Enter a number: "))
length = len(str(num))
revNum = ""
newNum = num

for i in range(length):
    revNum += str(newNum % 10)
    newNum = (newNum // 10)

revNum = int(revNum)

print(num, "is a palindrome!") if (num == revNum) else print(num,
   "and", revNum, "are not the same, so", num, "is not a
   palindrome.")
```

Listing 3: Problem 3

```
"""
 * ============================================================================
 *
 *        Filename:  HW1Prob3Brodskiy.py
 *        Assignment: Homework 1 Problem 3
 *        Title: Pythagorean Triple Finder
 *
 *    Description:  Prints all Pythagorean triples with side lengths less than 1000
 *
 *         Version:  1.0
 *         Created:  01/24/2023
 *        Revision:  N/A
 *          Python:  Python 3.9.2
 *
 *          Author:  M. Brodskiy
 *
 * ============================================================================
"""

for i in range(1, 1000):
    for j in range(i, 1000):
        for k in range(j, 1000):
            if (i ** 2 + j ** 2 == k ** 2):
                print(f"({i}, {j}, {k})")
```

Listing 4: Problem 4

```
"""
 * ============================================================
 *
 *        Filename:  HW1Prob4Brodskiy.py
 *        Assignment: Homework 1 Problem 4
 *        Title: Guessing Game
 *
 *    Description:  Randomly generates a number from 0-1000,
    having the player try to guess it with hints
 *
 *         Version:  1.0
 *         Created:  01/24/2023
 *        Revision:  N/A
 *          Python:  Python 3.9.2
 *
 *          Author:  M. Brodskiy
 *
 * ============================================================
"""

import random

num = int(1000 * random.random())
playAgain = -1

guess = int(input("Guess my number between 1 and 1000 with the
   fewest possible guesses: "))

while playAgain != 0:
    if guess > num:
        guess = int(input("Too high! Try again: "))
    elif guess < num:
        guess = int(input("Too low! Try again: "))
    else:
        print("Congratulations. You guessed the number!")
        num = int(1000 * random.random())
        playAgain = int(input("Play Again? Yes (1)/No (0):" ))
        if playAgain == 1:
            guess = int(input("Guess my number between 1 and 1000
               with the fewest possible guesses: "))
```

Listing 5: Problem 5

```
"""
 * ============================================================================
 *
 *        Filename:  HW1Prob5Brodskiy.py
 *        Assignment: Homework 1 Problem 5
 *        Title: Approximating Pi
 *
 *     Description:  Approximates pi using a Maclaurin series up to
     ten terms
 *
 *         Version:  1.0
 *         Created:  01/24/2023
 *        Revision:  N/A
 *          Python:  Python 3.9.2
 *
 *          Author:  M. Brodskiy
 *
 * ============================================================================
"""

sum = 1
cur_term = 1

for i in range(1, 11):
    print(f"{i}—Term: {4 * sum:.6f}")
    cur_term *= -((2 * i - 1) / (2 * i + 1))
    sum += cur_term
```

It takes 626 terms to achieve 3.14 consistently, 2,457 terms to achieve 3.141 consistently, and 146,601 terms to achieve 3.1415 consistently.