

Homework 1

Michael Brodskiy

Professor: M. Fanaei

January 24, 2023

Listing 1: Problem 1

```
1  """
2  * =====
3  *
4  *      Filename:  HW1Prob1Brodskiy.py
5  *      Assignment: Homework 1 Problem 1
6  *      Title: MPG Tracker
7  *
8  *      Description: Takes in miles driven and gallons used until a
9  *                   negative number is entered, then outputs an mpg value
10  *
11  *      Version:   1.0
12  *      Created:   01/24/2023
13  *      Revision:  N/A
14  *      Python:    Python 3.9.2
15  *
16  *      Author:    M. Brodskiy
17  * =====
18  """
19
20 galUsed = 0
21 milDriven = 0
22 sumMiles = 0
23 sumGal = 0
24
25 while 1 != 2:
26     galUsed = float(input("Enter the number of gallons used (any
27                             negative number to end): "))
28
29     if galUsed < 0:
30         break
31
32     milDriven = float(input("Enter the number of miles driven: "))
33     print(f"The miles/gallon for this tank was {milDriven /
34             galUsed:.6f}.")
35     sumMiles += milDriven
36     sumGal += galUsed
37
38 print(f"The overall average miles/gallon for the above tankfuls
39       was {sumMiles / sumGal:6f}.")
```

Listing 2: Problem 2

```
1  """
2  * =====
3  *
4  *      Filename:  HW1Prob2Brodskiy.py
5  *      Assignment: Homework 1 Problem 2
6  *      Title: Palindrome Verification
7  *
8  *      Description: Takes in a number and determines whether it is
9  *                  a palindrome
10 *
11 *      Version: 1.0
12 *      Created: 01/24/2023
13 *      Revision: N/A
14 *      Python: Python 3.9.2
15 *
16 *      Author: M. Brodskiy
17 * =====
18 """
19
20 num = int(input("Enter a number: "))
21 length = len(str(num))
22 revNum = ""
23 newNum = num
24
25 for i in range(length):
26     revNum += str(newNum % 10)
27     newNum = (newNum // 10)
28
29 revNum = int(revNum)
30
31 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

```
1  """
2  * =====
3  *
4  *      Filename:  HW1Prob3Brodskiy.py
5  *      Assignment: Homework 1 Problem 3
6  *      Title: Pythagorean Triple Finder
7  *
8  *      Description: Prints all Pythagorean triples with side
9  *                  lengths less than 1000
10 *
11 *      Version:   1.0
12 *      Created:   01/24/2023
13 *      Revision:  N/A
14 *      Python:    Python 3.9.2
15 *
16 *      Author:    M. Brodskiy
17 * =====
18 """
19
20 for i in range(1, 1000):
21     for j in range(i, 1000):
22         for k in range(j, 1000):
23             if (i ** 2 + j ** 2 == k ** 2):
24                 print(f"({i}, {j}, {k})")
```

Listing 4: Problem 4

```

1  """
2  * =====
3  *
4  *      Filename:  HW1Prob4Brodskiy.py
5  *      Assignment: Homework 1 Problem 4
6  *      Title:    Guessing Game
7  *
8  *      Description: Randomly generates a number from 0–1000,
9  *                  having the player try to guess it with hints
10 *
11 *      Version:   1.0
12 *      Created:   01/24/2023
13 *      Revision:  N/A
14 *      Python:    Python 3.9.2
15 *
16 *      Author:    M. Brodskiy
17 * =====
18 """
19
20 import random
21
22 num = int(1000 * random.random())
23 playAgain = -1
24
25 guess = int(input("Guess my number between 1 and 1000 with the
26                  fewest possible guesses: "))
27
28 while playAgain != 0:
29     if guess > num:
30         guess = int(input("Too high! Try again: "))
31     elif guess < num:
32         guess = int(input("Too low! Try again: "))
33     else:
34         print("Congratulations. You guessed the number!")
35         num = int(1000 * random.random())
36         playAgain = int(input("Play Again? Yes (1)/No (0):" ))
37         if playAgain == 1:
38             guess = int(input("Guess my number between 1 and 1000
39                             with the fewest possible guesses: "))

```

Listing 5: Problem 5

```
1  """
2  * =====
3  *
4  *      Filename:  HW1Prob5Brodskiy.py
5  *      Assignment: Homework 1 Problem 5
6  *      Title:    Approximating Pi
7  *
8  *      Description:  Approximates pi using a Maclaurin series up to
9  *                   ten terms
10 *
11 *      Version:    1.0
12 *      Created:    01/24/2023
13 *      Revision:   N/A
14 *      Python:     Python 3.9.2
15 *
16 *      Author:     M. Brodskiy
17 * =====
18 """
19
20 sum = 1
21 cur_term = 1
22
23 for i in range(1, 11):
24     print(f"{i}-Term: {4 * sum:.6f}")
25     cur_term *= -((2 * i - 1) / (2 * i + 1))
26     sum += cur_term
```