Calcutta Institute of Engineering and Management
Department of Information Technology

IT Workshop (Python) - PCC-CS393

NAME:	DIBBYARUP DUTTA
ROLL NO.:	16500221005
ASSIGNMENT:	WEEK 3
DATE OF ASSIGNMENT:	9th August, 2022
DATE OF SUBMISSION:	

_____
Signature of the Faculty with Date

Question-1

Create a python program that computes the average of a collection of values entered by the user. The user will enter 0 as a sentinel value to indicate that no further values will be provided. Your program should display an appropriate error message if the first value entered by the user is 0.

Keep in mind that the 0 at the end of the input should not be included in the average.

CODE:

```python
Sum, count = 0, 0
print("Enter numbers: ")
while True:
    value = int(input())
    if value == 0:
        if count == 0:
            print("Error: First number is 0")
            exit()
        break;
    sum += value
    count += 1
average = sum/count
print(f"Average = {average}")
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python average.py
Enter numbers:
8
9
4
7
1
0
Average = 5.8
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python average.py
Enter numbers:
0
```

Question-2

A particular zoo determines the price of admission based on the age of the guest. Guests 2 years of age and less are admitted without charge. Children between 3 and 12 years of age cost $14.00. Seniors aged 65 years and over cost $18.00. Admission for all other guests is $23.00.

Create a program that begins by reading the ages of all of the guests in a group from the user, with one age entered on each line. The user will enter a blank line to indicate that there are no more guests in the group. Then your program should display the admission cost for the group with an appropriate message. The cost should be displayed using two decimal places.

CODE:

```python
total_cost = 0
print("Enter the age of the guests: ")
while True:
    age = input()
    if age == '': break
    age = int(age)
    if age<=2:
        total_cost += 0
    elif 3<=age<=12:
        total_cost += 14
    elif age>=65:
        total_cost += 18
    else:
        total_cost += 23
print(f"Total Admission Cost: ${total_cost}.00")
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python zoo.py
Enter the age of the guests:
7
5
65
2
37
42

Total Admission Cost: $92.00
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python zoo.py
Enter the age of the guests:

Total Admission Cost: $0.00
```

Question-3

The greatest common divisor of two positive integers, n and m, is the largest number, d, which divides evenly into both n and m. There are several algorithms that can be used to solve this problem, including:

```
Initialize d to the smaller of m and n.
While
        d does not evenly divide m or d does not evenly divide n
do
        Decrease the value of d by 1
Report d as the greatest common divisor of n and m
```

Write a program that reads two positive integers from the user and uses this algorithm to determine and report their greatest common divisor.

CODE:

```python
def gcd(m, n):
    d = n if m>n else m
    while m%d!=0 or n%d!=0:
        d -= 1
    print(f"Greatest Common Divisor: {d}")
m, n = input("Enter two numbers: ").split()
gcd(int(m), int(n))
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python findgcd.py
Enter two numbers: 65 130
Greatest Common Divisor: 65
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python findgcd.py
Enter two numbers: 1333 237
Greatest Common Divisor: 1
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python findgcd.py
Enter two numbers: 52 143
Greatest Common Divisor: 13
```

Question-4

The prime factorization of an integer, n, can be determined using the following steps:

```
Initialize factor to 2
While factor is less than or equal to n do
        If
                n is evenly divisible by factor then Conclude that factor
                is a factor of n.
                Divide n by factor using floor division
        Else
                Increase factor by 1
```

Write a program that reads an integer from the user. If the value entered by the user is less than 2 then your program should display an appropriate error message. Otherwise your program should display the prime numbers that can be multiplied together to compute n, with one factor appearing on each line.

CODE:

```python
def primeFactorization(num):
    factor = 2
    print(f"The prime factors of {num} are:")
    while factor<=num:
        if num%factor==0:
            print(factor)
            num /= factor
        else:
            factor += 1

num = int(input("Enter a number: "))
if num<2:
    print("Entered number is less than 2.")
else:
    primeFactorization(num)
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python primefactor.py
Enter a number: 143
The prime factors of 143 are:
11
13
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python primefactor.py
Enter a number: 72
The prime factors of 72 are:
2
2
2
3
3
```

Bonus Question-1

Consider a sequence of integers that is constructed in the following manner:

Start with any positive integer as the only term in the sequence
While

the last term in the sequence is not equal to 1

do

If the last term is even then

Add another term to the sequence by dividing the last term by 2 using floor division.

Else

Add another term to the sequence by multiplying the last term by 3 and adding 1.

The Collatz conjecture states that this sequence will eventually end with one when it begins with any positive integer. Although this conjecture has never been proved, it appears to be true.
Create a program that reads an integer, n, from the user and reports all of the values in the sequence starting with n and ending with one. Your program should allow the user to continue entering new n values (and your program should continue displaying the sequences) until the user enters a value for n that is less than or equal to zero.

CODE:

```python
def dispSeq(num):
    while num != 1:
        print(num, end=" ")
        if num&1 == 0:
            num = num//2
        else:
            num = (num*3)+1
    print(num)

print("Enter numbers: ")
while True:
    num = int(input())
    if num <= 0: exit()
    dispSeq(num)
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python sequence.py
Enter numbers:
8
8 4 2 1
13
13 40 20 10 5 16 8 4 2 1
17
17 52 26 13 40 20 10 5 16 8 4 2 1
0
```

Bonus Question-2

What's the minimum number of times you have to flip a coin before you can have three consecutive flips that result in the same outcome (either all three are heads or all three are tails)? What's the maximum number of flips that might be needed? How many flips are needed on average? In this exercise we will explore these questions by creating a program that simulates several series ofcoin flips.

Create a program that uses Python's random number generator to simulate flipping a coin several times. The simulated coin should be fair, meaning that the probability of heads is equal to the probability of tails. Your program should flip simulated coins until either 3 consecutive heads of 3 consecutive tails occur. Display an H each time the outcome is heads, and a T each time the outcome is tails, with all of the outcomes for one simulation on the same line. Then display the number of flips that were needed to reach 3 consecutive occurrences of the same outcome. When your program is run it should perform the simulation 10 times and report the average number of flips needed.

CODE:

```python
import random

def toss():
    Hcount, Tcount = 0, 0
    count = 0
    while Hcount!=3 and Tcount!=3:
        count += 1
        val = random.randint(0,1)
        if val == 0:
            print("H", end=" ")
            Hcount += 1
            Tcount = 0
        else:
            print("T", end=" ")
            Tcount += 1
            Hcount = 0
    print(f"({count} flips)", end="")
    return count

sum = 0
for i in range(10):
    sum += toss()
    print()

print(f"On average, {sum/10} flips were needed.")
```

OUTPUT:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python flips.py
H T H H T T T (7 flips)
T T H T H T H T T T (10 flips)
H T T H H T T T (8 flips)
T H H T H H H (7 flips)
T T H H T H H T T T (10 flips)
T H H T H T T T (8 flips)
T H H T T T (6 flips)
T H H T H H T T T (9 flips)
T T H H H (5 flips)
H T H T T T (6 flips)
On average, 7.6 flips were needed.
PS F:\COM.PROG.DOCS\CODE FILES\Lang\Python\Week 3> python flips.py
T H T H T H H H (8 flips)
T T T (3 flips)
H T H T H H T T T (9 flips)
T T H T T H T H H H T H H H (13 flips)
H T H T T H T H T H H T T H T T T (15 flips)
T T H T T T (6 flips)
T T H T T T (6 flips)
T T H H T T T (7 flips)
H T H T T T (6 flips)
H T H T T H H H (8 flips)
On average, 8.1 flips were needed.
```