**Fundamentals of Data Science**

**The British College**

**Bi-weekly Assignment Solutions (Week 3 & 4)**

**Student Name:** Praful Tiwari  
**Module:** Fundamentals of Data Science  
**Module Code:** UFCFK1-15-0  
**Tutor:** Saurav Gautam  
**Academic Year:** 2025

### 1. Function for Arithmetic Operations

**Description:** A function that accepts two numbers and displays sum, difference, product, and remainder.

# Function to perform arithmetic operations  
def arithmetic\_operations(a, b):  
 print("Sum:", a + b)  
 print("Difference:", a - b)  
 print("Product:", a \* b)  
 print("Remainder:", a % b)  
  
# Example call  
arithmetic\_operations(10, 3)

A black screen with white text

Description automatically generated

### 2. Separate Functions for Arithmetic Operations

**Description:** Each operation is handled by a separate function and returns values.

def add(a, b):  
 return a + b  
  
def subtract(a, b):  
 return a - b  
  
def multiply(a, b):  
 return a \* b  
  
def divide(a, b):  
 return a / b  
  
def modulo(a, b):  
 return a % b  
  
def floor\_div(a, b):  
 return a // b  
  
# Example usage:  
a = 10  
b = 4  
print("Addition:", add(a, b))  
print("Subtraction:", subtract(a, b))  
print("Multiplication:", multiply(a, b))  
print("Division:", divide(a, b))  
print("Modulo:", modulo(a, b))  
print("Floor Division:", floor\_div(a, b))

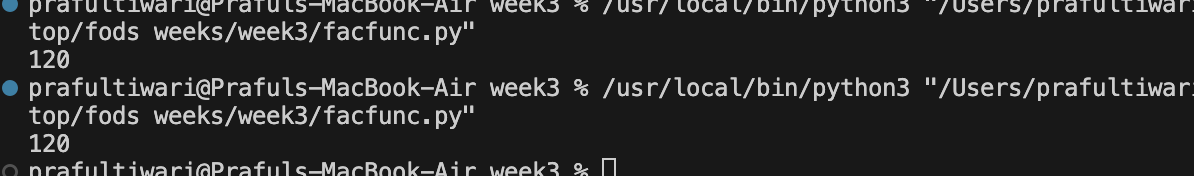
A black screen with white text

Description automatically generated

### 3. Function for Factorial

**Description:** Returns factorial of a given number.

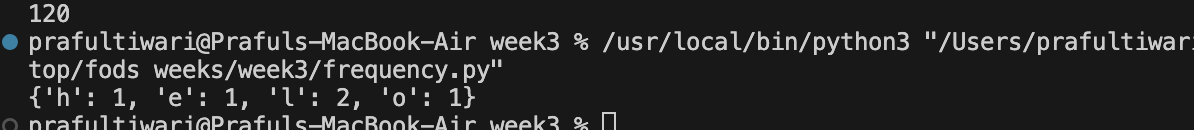
def factorial(n):  
 result = 1  
 for i in range(1, n + 1):  
 result \*= i  
 return result  
  
print("Factorial:", factorial(5))



### 4. Frequency of Each Number in List

**Description:** Counts frequency of each number in a list.

def count\_frequency(lst):  
 freq = {}  
 for num in lst:  
 freq[num] = freq.get(num, 0) + 1  
 return freq  
  
print(count\_frequency([1, 2, 2, 3, 3, 3]))



### 5. Sorted Order of Names

**Description:** Sorts a list of names.

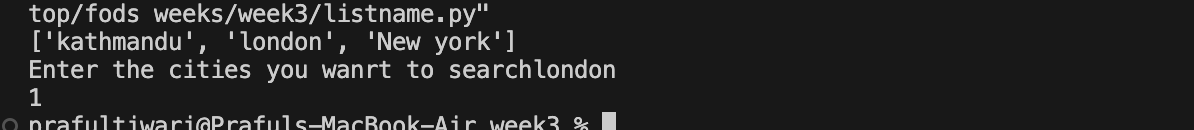
def sort\_names(names):  
 return sorted(names)  
  
print(sort\_names(["John", "Alice", "Bob"]))



### 6. Search City in List

**Description:** Returns index of city if found.

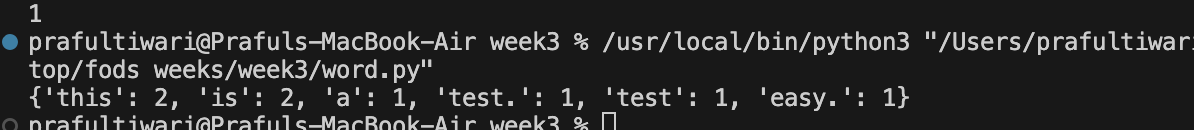
def search\_city(city\_list, target):  
 if target in city\_list:  
 return city\_list.index(target)  
 else:  
 return "City not found."  
  
print(search\_city(["Kathmandu", "Pokhara"], "Pokhara"))



### 7. Word Frequency in Sentence

**Description:** Returns dictionary of word frequencies.

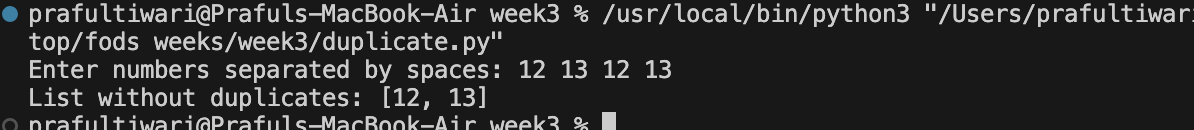
def word\_frequency(sentence):  
 words = sentence.lower().split()  
 freq = {}  
 for word in words:  
 freq[word] = freq.get(word, 0) + 1  
 return freq  
  
print(word\_frequency("This is a test. This test is easy."))



### 8. Remove Duplicate from List

**Description:** Returns list without duplicate values.

def remove\_duplicates(lst):  
 return list(set(lst))  
  
print(remove\_duplicates([1, 2, 2, 3, 4, 4]))



### 9. Book Details Dictionary

**Description:** Stores 5 books with details in dictionary.

books = {}  
  
for i in range(5):  
 title = input("Enter title: ")  
 author = input("Enter author: ")  
 isbn = input("Enter ISBN: ")  
 cost = float(input("Enter cost: "))  
 books[isbn] = {"title": title, "author": author, "cost": cost}  
  
for isbn, info in books.items():  
 print(f"ISBN: {isbn}, Title: {info['title']}, Author: {info['author']}, Cost: {info['cost']}")

A black background with white text

Description automatically generated

### 10. Segregate Even and Odd Numbers

**Description:** Takes continuous input and separates even/odd.

even = []  
odd = []  
  
while True:  
 num = input("Enter a number (or type 'exit'): ")  
 if num.lower() == 'exit':  
 break  
 if num.isdigit():  
 n = int(num)  
 if n % 2 == 0:  
 even.append(n)  
 else:  
 odd.append(n)  
  
print("Even numbers:", even)  
print("Odd numbers:", odd)

A screen shot of a computer

Description automatically generated

### 11. Card Guessing Game

**Description:** Game to guess card value and suit.

import random  
  
values = ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack', 'Queen', 'King', 'Ace']  
suits = ['Heart', 'Diamond', 'Club', 'Spades']  
  
selected\_value = random.choice(values)  
selected\_suit = random.choice(suits)  
  
print("Guess the card!")  
value\_guess = input("Enter card value: ")  
suit\_guess = input("Enter card suit: ")  
  
if value\_guess == selected\_value and suit\_guess == selected\_suit:  
 print("♥ ☺ Perfect Guess!")  
elif value\_guess == selected\_value or suit\_guess == selected\_suit:  
 print(":) Partial Match")  
else:  
 print("❤ Game Over")

A screenshot of a computer

Description automatically generated

**End of Assignment 2 Solution**