Qn1. Program1:

#1. Write a function that accepts a string and calculate the number of upper case letters and lower case letters.

def calci (word):

    #Uppercase and lower case count

    lower\_case = 0

    upper\_case = 0

    for ch in word:

        if ch.isupper(): #Checkes whether the character is in uppercase or not

            upper\_case+=1  #if it is it adds by 1

        if ch.islower(): #Checks whether the character is in lowercase or not

            lower\_case+=1 #if it is adds by 1

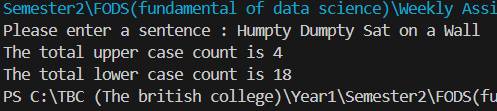
    print (f"The total upper case count is {upper\_case}")

    print (f"The total lower case count is {lower\_case}")

words = input("Please enter a sentence : ")

calci(words)

Output:



Qn2. Program2:

#2. Write a function to check whether the given number is prime or not.

def prime (num):

    i=1

    count = 0 #how many time a number can be divided

    if num <=1:

        return False

    for i in range (i,num+1):

        if num%i==0:

            count +=1

    if count >2: #if num is divided by number more than itself and 1

        return False

    return True

number = int(input("Please Enter a number: "))

#to check

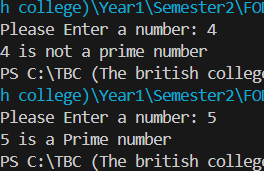
if prime(number):

    print(f"{number} is a Prime number ")

else:

    print(f"{number} is not a prime number")

Output:



Qn3. Program3:

#3. Write a function to check whether the given number is Armstrong or not.

def arm\_num (num):

    a = num

    sum = 0

    length = len(str(num)) #to find the length

    while (a>0):

        rem = a%10

        sum = sum + pow(rem, length)

        a= a//10

    return sum==num #Checks is sum and number are equal or not

number=int(input("Enter a number : "))

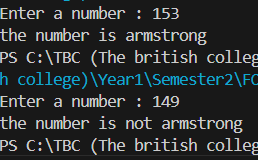
if arm\_num(number)==False:

    print("the number is not armstrong")

elif arm\_num(number)==True:

    print("the number is armstrong")

Output:



Qn4. Program 4:

#4. Write a function to accept a list of names and return the sorted order of names back.

def lst\_name (name):

    name\_lst= name.split(",") #Converts the string into a list whenever comma appears

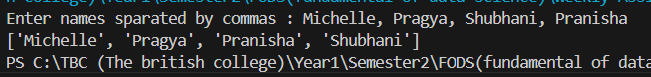
    name\_lst= [name.strip() for name in name\_lst] #removes whitespace

#strip() method is applied to all names in the list before sorting them

    return sorted(name\_lst)

names= input ("Enter names separated by commas : ")

print(lst\_name(names))

Output:

Qn5. Program 5:

#5. Create a program called calculator with functions to perform the following arithmetic calculations,

#  each should take two decimal parameters and return the result of the arithmetic calculation in question.

# A. Addition

# B. Subtraction

# C. Multiplication

# D. Division

# E. Truncated division

# F. Modulus

# G. Exponentiation

def addition (a,b):

    add = a + b

    return add

def subtraction (a,b):

    sub = a - b

    return sub

def multiplication (a, b):

    multi = a \* b

    return multi

def division (a,b):

    quotient = a/b

    quotient = round(quotient,2)

    return quotient

def Truncated\_div (a,b):

    quotient = a//b #doesnot returns decimal point

    return quotient

def modulus (a,b):

    rem = a%b

    return rem

def exponential (a,b):

    expo = pow(a,b)

    return expo

num1 = int (input("Enter the first number : "))

num2 = int (input("Enter the second number : "))

print (f"The addition of {num1} and {num2} is {addition(num1,num2)}")

print (f"The subtraction of {num1} and {num2} is {subtraction(num1,num2)}")

print (f"The multiplication of {num1} and {num2} is {multiplication(num1,num2)}")

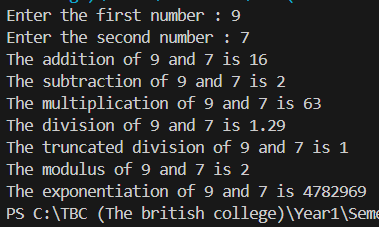
print (f"The division of {num1} and {num2} is {division(num1,num2)}")

print (f"The truncated division of {num1} and {num2} is {Truncated\_div(num1,num2)}")

print (f"The modulus of {num1} and {num2} is {modulus(num1,num2)}")

print (f"The exponentiation of {num1} and {num2} is {exponential(num1,num2)}")

Output:



Qn6. Program6 :

#6. Write a program that prompts the user for a series of integers and stores in a list only the values between 1-100,

#  and displays the resulting list.

num = input ("Enter the numbers separated by comma : ")

number =num.split(",") #splits the numbers into each element

numbers =[]

for nums in number:

    #to check the input is number or alpha

    if nums.isdigit():

        i = int(nums)

        #to check whether num is from 1-100 or not

        if i >=1 and i<=100:

            numbers.append(i)

        else:

            print ("Please enter a number from 1-100")

    else:

        print (f"{nums} is not a valid integer.")

print (f"The numbers between 1-100 are {numbers}")

Output:

Qn7. Program7:

#7 Write a program that prompts the user to enter a list of names and store them in a list.

# The program should display how many times the letter 'a appears within the list.

name\_List = []

name = input("Enter the names separated by comma : ")

names= name.split(",") #Separating the name by comma to make a list

for word in names:

    name\_List.append(word) #appending each name in list

#Checking how many times a appeared

a\_count = 0

for a in name\_List:

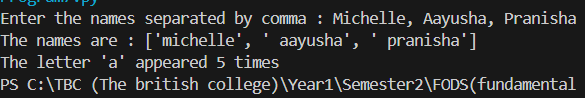
    a\_count += a.lower().count('a')

#OR #a\_count= sum(word.a().count('a')for a in name\_List)

print (f"The names are : {name\_List}")

print (f"The letter 'a' appeared {a\_count} times ")

Output:



Qn8. Program 8:

#8. Write a program that prompts the user to enter integer values to populate two lists,

# then prints messages to determine the following:

# (a) Whether the lists are of the same length.

# (b) Whether the elements in each list sum to the same value.

# (c) Whether there are any values that occur in both lists

lst1 = input (" Enter the elements of list 1 separated by comma ").split(",")

#Spliting the input element into individual number

lst2 = input (" Enter the elements of list 2 separated by comma ").split(",")

num1= [int(num) for num in lst1] #converting each element into integer

num2= [int(num) for num in lst2]

print (f"List 1: {num1} \n List 2: {num2}")

if len(num1)== len(num2):

  print (f"List 1 and List 2 have the same length.That is : {len(num1)}")

else:

  print (f"List 1 length : {len(num1)} \nList 2 length : {len(num2)}")

sum\_num1 = 0

sum\_num2 = 0

for num in num1: #adding the value of list 1

  sum\_num1+=num

for num in num2: #adding the value of list 2

  sum\_num2+=num

if sum\_num1 == sum\_num2:

  print (f"The sum of List1 and the sum of List 2 are equal : {sum\_num1}")

else:

  print (f"The sum of List 1 : {sum\_num1} \nThe sum of List 2 : {sum\_num2}")

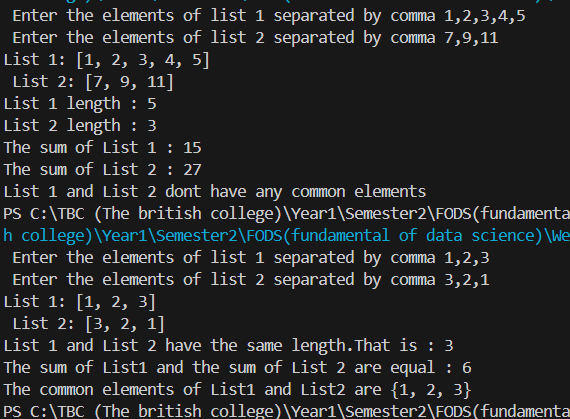
intersection = set(num1) & set(num2) #using logical AND to check if there are common numbers or not

if intersection:

  print (f"The common elements of List1 and List2 are {intersection}")

else:

  print ("List 1 and List 2 dont have any common elements")

Output:

Qn9. Program9:

Output:

#9. Write a function called add\_daily\_temp that is given a (possibly empty) dictionary meant to hold the average daily temperature

#  for each day of the week, a temperature value, and the day of the week for the recorded temperature.

# The function should then add the temperature to the dictionary only if it does not already contain a temperature for that day.

#  The function should return the resulting dictionary, whether it is updated or not.

#Creating function with 3 parameters

def add\_daily\_temp(temp\_dict, temperature, day):

    if day not in temp\_dict:

        temp\_dict[day] = [temperature] #add the particular day to temp\_dict

    return temp\_dict #return the updated temp\_dict

#Dictionary with some initial temperature value

temp\_week = {'Sunday' : 20,'Tuesday':21, 'Thursday':22 }

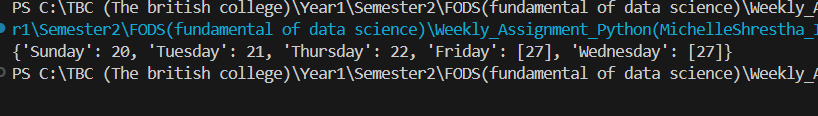
temp\_week = add\_daily\_temp(temp\_week,26,'Sunday') #passing the arguments

temp\_week = add\_daily\_temp(temp\_week,27,'Friday')

temp\_week = add\_daily\_temp(temp\_week,27,'Wednesday')

print(temp\_week)

Output:



Qn10. Program10:

#.10.  Write a function named get\_daily\_temps that prompts the user for the average temperature

# for each day of the week and returns a dictionary containing the information the user entered.

def daily\_temp():

  temperature = {}#creating empty dictionary to store values later

  i = 0

  for i in range (7):#iterates 7 times 0-6

    day = input ("Enter the day of the week : ").capitalize()

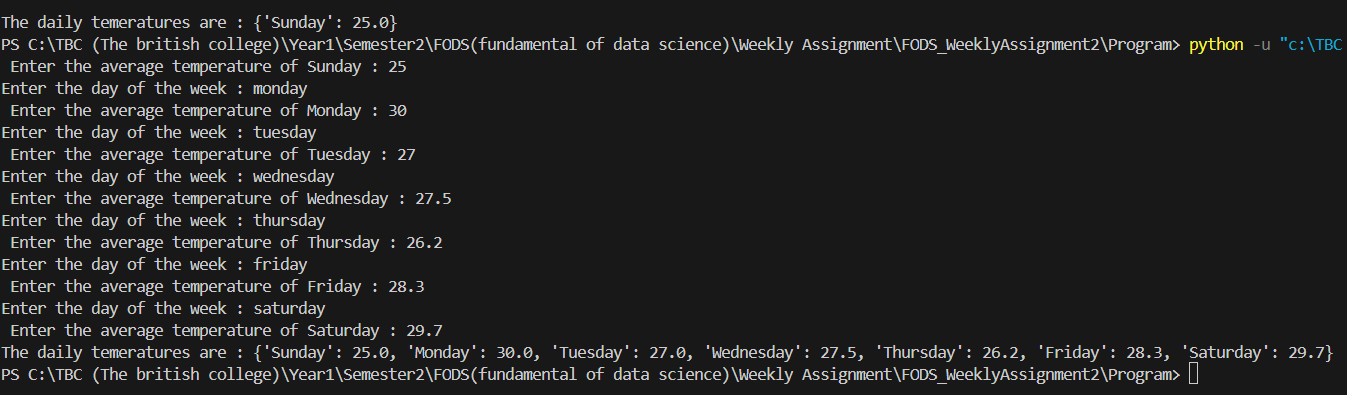
    temp = float(input(f" Enter the average temperature of {day} : "))

    temperature[day]= temp # this line indicates day as a key and temp as its vaule in temperature dictionary

  return temperature

temp = daily\_temp()

print (f"The daily temeratures are : {temp}")

Output:

Qn11. Program11:

#11. Create three dictionaries:

#dic1 = {1:10, 2:20}

#dic2 = {3:30, 4:40}

#dic3 = {5:50, 6:60}

#(a) Write code to concatenate these dictionaries to create a new one.

#  Create a variable called nums to store the resulting dictionary.

#(b) Write code to add a new key/value pair to the dictionary nums: (7, 70)

#(c) Write code to update the value of the item with key 3 in nums to 80

#(d) Write code to remove the third item from dictionary nums.

#(e) Write code to sum all the items in the dictionary nums

#(f) Write code to multiply all the items in the dictionary nums

#(g) Write code to retrieve the maximum and minimum values in nums

dic1 = {1:10, 2:20}

dic2 = {3:30, 4:40}

dic3 = {5:50, 6:60}

nums= {} #empty dictionaries

nums.update(dic1) #adds items

nums.update(dic2)

nums.update(dic3)

print(f"The concatenated dictionary: {nums}")

nums[7]= 70 #adding new key and value in nums dictionary

print (f"Dictionary after adding new key and value: {nums}")

nums[3]= 80 #updating the key 3 value to 80

print(f"Dictionary after updating the key 3 value to 80: {nums}")

nums.pop(3)

print (f"Dictionary after removing the third item: {nums}")

total\_sum = 0

for value in nums.values(): #adding the only values not keys

  total\_sum+=value

print(f"The total sum of all items: {total\_sum}")

total\_multiple = 1

for value in nums.values(): #adding the values only not keys

  total\_multiple\*=value

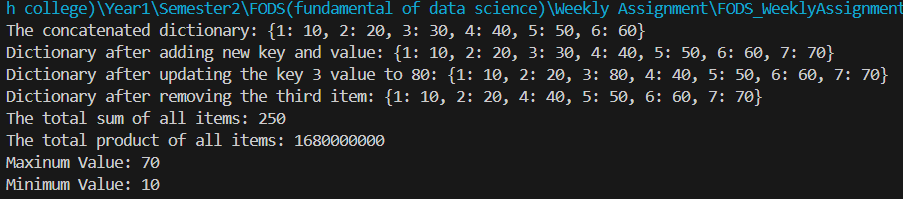
print(f"The total product of all items: {total\_multiple}")

maximum\_value = max(nums.values())

minimum\_value = min(nums.values())

print (f"Maxinum Value: {maximum\_value}\nMinimum Value: {minimum\_value}")

Output:



Qn12. Program12:

#12. Create two sets:

#set1 = {20, 40, 60}

#set2 = {10, 20, 30, 40, 50, 60}

#(a) Write code to perform a union of these sets. Print the length of the resulting set.

#(b) Write code to perform an intersection of set1 and set2.

#(c) Write code to compute the symmetric difference between set1 and set2

#(d) Write code to add the value 40 to set1, did the set change?

#(e) Write code to remove value 20 from set2

set1 = {20,40,60}

set2 = {10,20,30,40,50,60}

union\_set = set1.union(set2) #can also use set1 | set2 ("|" is an union operator)

union\_set\_length = len(union\_set)

print(f"The union of set 1 and set2: {union\_set}")

print(f"Length of the union set: {union\_set\_length}")

intersection\_set = set1 & set2 # "&" is an intersection operator

print (f"The intersection of set1 and set2: {intersection\_set}")

symmetric\_difference\_set = set1 ^ set2 # "^" is a symmetric difference operator

print(f"The symmetric difference between set1 and set2: {symmetric\_difference\_set}")

set1.add(40)

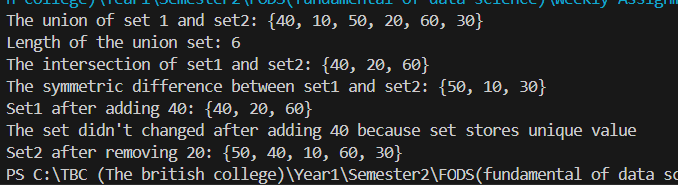
print(f"Set1 after adding 40: {set1}")

print("The set didn't changed after adding 40 because set stores unique value")

set2.remove(20)

print(f"Set2 after removing 20: {set2}")

Output:



Qn13. Program 13:

#13. Create a function called word\_intersection that prompts the user for two English words,

# and displays which letters the two words have in common.

def word\_intersection():

  word1= input("Enter the first english word: ").lower() #lower is used to make case-insensitive

  word2= input("Enter the second english word: ").lower()

  common\_letters=""#creating empty string

  for letter in word1:

    #Checking for unique common letters

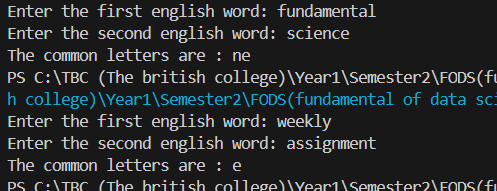
    if letter in word2 and letter not in common\_letters:

      common\_letters+=letter

  return common\_letters

common\_letter = word\_intersection()

print (f"The common letters are : {common\_letter}")

Output: