INDEX

Module	Programs in the Module	From Page	To Page	Date/s	Marks	Signature
Programs on Tuples	Write a Python script to create a tuple with different data types.					
	Write a Python script to find the repeated items of a tuple.					
	Write a Python script to replace last value of tuples in a list.					
	Write a Python script to sort a tuple by its float element.					
Programs on Sets and Dictionaries	Write a Python script to add member(s) in a set.					
	Write a Python script to perform Union, Intersection, difference, and symmetric difference of given two sets.					
	Write Python script to test whether every element in S is in T and every element in T is in S.					
	Write a Python script to sort (ascending and descending) a dictionary by value.					
	Write a Python script to check whether a given key already exists or not in a dictionary.					

	Write a Python script to concatenate following dictionaries to create a newone.
	Sample Dictionary:
	dic1 = {1:10, 2:20}
	dic2 = {3:30, 4:40}
	dic3 = {5:50,6:60}
	Expected Result:
	{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}					

	Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.					
	Write a Python program to map two lists into a dictionary.					

Module	Programs in the Module	From Page	To Page	Date/s	Marks	Signature
Programs on functions and recursion	Define a function max_of_three() that takes three numbers as arguments and returns the largest of them.					
	Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between given range X and Y.					
	Define functions to find mean, median, mode for the given numbers in a list.					
	Define a function which generates Fibonacci series upto 'n' numbers.					
	Implement a python script for factorial of number by using recursion.					
	Implement a python script to find GCD of given two numbers using recursion.					
Programs on Strings	Implement Python Script to perform various operations on string using string libraries.					
	Implement Python Script to check given string is palindrome or not.					
	Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.					
	Implement python script that takes a list of words and returns the length of the longest one.					

Module	Programs in the Module	From Page	To Page	Date/s	Marks	Signature
Programs on Regular Expressions	Write a Python script to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).					
	Write a Python script to check whether password is valid or not. Conditions for a valid password are: 1. Should have at least one number. 2. Should have at least one uppercase and one lowercase character. 3. Should have at least one special symbol. 4. Should be between 6 to 20 characters long.					
Programs on Python Libraries – Matplotlib	Write a Python program to draw a line with suitable label in the x axis, y axis and a title.					
	Write a Python program to plot two or more lines with legends, different widths and colors.					
	Write a Python program to create multiple plots.					

	Write a Python programming to display a bar chart using different color for each bar.					
	Write a Python programming to create a pie chart with a title					
	Write a Python program to draw a scatter plot with empty circles taking a random distribution in X and Y and plotted against each other					

Average of the Record Marks: ________________

Signature of the Faculty Incharge

Introduction: Language Basics and Example Problems

a) Implement Python Script for checking the given year is leap year or not.

Program:

```
year = int(input(”Enter a year: "))
if (year % 4) ==  0:
  if (year % 100) ==  0:
     if (year % 400) == 0:
        print(year, "is a leap year")
     else:
        print(year, "is not a leap year")
  else:
     print(year, "is a leap year")
else:
  print(year, "is not a leap year")
```

Output:

b) Implement Python Script for finding biggest number among 3 numbers.

Program:

```
a = int(input("Enter Number 1: "))
b = int(input(”Enter Number 2: ”))
c = int(input("Enter Number 3: "))

if a>b and a>c:
   print(a, "is biggest among the three numbers")
elif b>c:
   print(b, "is biggest among the three numbers")
else:
   print(c, "is biggest among the three numbers")
```

Output:

c) Implement Python Script for displaying reversal of a number.

Program:

```
number = int(input("Enter the integer number: ”))

revs_number = 0

while (number > 0):
    remainder = number % 10
    revs_number = (revs_number * 10) + remainder
    number = number // 10

print("The reverse number is : ”, revs_number)
```

Output:

d) Implement Python Script to check given number is Armstrong or not.

Program:

```
num = int(input(”Enter an Integer Number: ”))
no_of_digits = len(str(num))
sum = 0
temp = num

while temp > 0:
  digit = temp % 10
  sum += digit ** no_of_digits
  temp //= 10

if num == sum:
  print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

Output:

e) Implement Python Script to print sum of N natural numbers.

Program:

```
num = int(input(”Enter an Integer to print sum of numbers up to num: "))

if num < 0:
  print("Enter a positive number")
else:
  sum = 0
  while(num > 0):
     sum += num
     num -= 1
  print("The sum is", sum)
```

Output:

f) Implement Python Script to check given number is palindrome or not.

Program:

```
number = int(input("Enter any number : "))

temp=number
reverse_num=0

while(number>0):
   digit=number%10
   reverse_num=reverse_num*10+digit
   number=number//10

if(temp==reverse_num):
   print(”The number is palindrome!")
else:
   print(”Not a palindrome!”)
```

Output:

g) Implement Python script to print factorial of a number.

Program:

```
num = int(input(”Enter a Number to find Factorial: ”))
factorial = 1

if num < 0:
  print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
  print("The factorial of 0 is 1")
else:
  for i in range(1,num + 1):
     factorial = factoria1*i
  print("The factorial of ",num," is ",factorial)
```

Output:

h) Implement Python Script to print all prime numbers within the given range.

Program:

```
lower_range = int(input("Enter Lower Range: ”))
upper_range = int(input("Enter Upper Range: "))

for num in range(1ower_range,upper_range + 1):
  if num > 1:
    for i in range(2, num):
      if (num % i) == 0:
        break
    else:
      print(num, end=”,”)
```

Output:

i) Implement Python Script to calculate the series: $S=1+x+x^2+x^3+........ x$•

Program:

```
sum = 1
print("Here we will calculate the sum of series 1+x+x^2+. ... +x^n")
x=int(input(”Enter the value of x: "))
n = int(input(”Enter the value of n: "))
for i in range(1,n+1):
   sum = sum +(x**i)
print("The sum of the series is:", sum)
```

Output:

j) Implement Python Script to print the following pattern:

```
  *
 * *
* * *
```

Program:

```
n = int(input(”Enter a Number to build Pyramid: ”))
k = n - 1

for i in range(0, n):
   for j in range(0, k):
      print(end=" ”)
   k = k - 1
   for j in range(0, i+1):
      print("* ", end="")
   print(”\r")
```

Output:

Module 1: Exercise Programs on Lists

a) Write a Python script to display elements of list in reverse order.

Program:

```
systems = ['Windows', 'macos', 'Linux']

print('Original List: ’, systems)

systems.reverse()

print('Updated List: ', systems)
```

Output:

b) Writea Python script to findthe minimum andmaximum elements without using built-in operations in the lists.

Program:

```
 '   []
n = int(input("How many elements you want to enter in a list? ”))

for i in range(0, n):
   a = int(input(”Enter a Number to be added to list: ”))
   x.append(a)

max_val = x[0]
min_val = x[0]

for check in x:
   if check > max_val:
      max_val = check

for check in x:
   if check < min_val:
      min_val = check


print("Maximum Number in the List is: ", max_val)
print("Minimum Number in the List is:", min_val)
```

Output:

c) Write a Python script to remove duplicates from a list.

Program:

```
x = []

n = int(input(”How many elements you want to enter in a list? "))

for i in range(0, n):
   a = int(input("Enter a Number to be added to list: "))
   x.append(a)

new_list = []

for num in x:
   if num not in new_list:
      new_list.append(num)

print(new_list)
```

Output:

d) Write a Python script to append a list to the second list.

Program:

```
 '   []
m = int(input("How many elements you want to enter in a list x? "))
for i in range(0, m):
   a = int(input("Enter a Number to be added to list x: "))
   x.append(a)

n = int(input(”How many elements you want to enter in a list y? "))
y = []
for i in range(0, n):
   b = int(input(”Enter a Number to be added to list y: ”))
   y.append(b)

new_list = x + y

print(new_list)
```

Output:

e) Write a Python script to count the number of strings in a list where the string length is 2 ormore.

Program:

```
 '   []
count = 0
m = int(input("How many strings you want to enter in a list x? "))
for i in range(0, m):
   a = input("Enter a String to be added to list x: ”)
   x.append(a)
   if len(x[i]) >= 2:
      count = count + 1

print("Total Strings whose length is >2 in the List are: ”, count)
```

Output:

Module 2: Exercise Programs on Tuples

a) Write a Python script to create a tuple with different data types.

Program:

```
tuplex = ("tuple", False, 3.2, 1)
print(tuplex)
```

Output:

b) Write a Python script to find the repeated items of a tuple.

Program:

```
tuplex = 2, 4, 5, 6, 2, 3, 4, 4, 7
print(tuplex)
count = tuplex.count(4)
print(count)
```

Output:

c) Write a Python script to replace last value of tuples in a list.

Program:

```
l = [(10, 20, 40), (40, 50, 60), (70, 80, 90)]
print([t[:-1] + (100,) for t in l])
```

Output:

d) Write a Python script to sort a tuple by its float element.

Program:

```
price = [('item1', '12.20'), ('item2', '15.10'), ('item3',  '24.5')]
print( sorted(price, key=lambda x: float(x[1]), reverse=True))
```

Output:

Module 3 & 4:

Exercise Programs on Sets and Dictionaries

a) Write a Python script toadd member(s) in a set.

Program:

```
color_set = set()
color_set.add("Red")
color_set.update(["Blue", ”Green"])
print(color_set)
```

Output:

b) Write a Python script to perform Union, Intersection, difference and symmetric difference of given two sets.

Program:

```
A = {0, 2, 4, 6, 8}
B = {1, 2, 3, 4, 5}
print("Union :”, A | B)
print("Intersection :", A & B)
print("Difference :", A - B)
print("Symmetric difference :", A ^ B)
```

Output:

c) Write Python script to test whether every element in S is in T and every element in T is in S.

Program:

```
S = set([”apple", ”mango"])
T = set(["mango", "banana"])
print(S <= T)
print(S >= T)
print(S == T)
print(S != T)
```

Output:

d) Write a Python script to sort (ascending and descending) a dictionaryby value.

Program:

```
dt = {8:4, 1:6, 6:3}

asc = {key: value for key, value in sorted(dt.items(), key=lambda item:item[1])}

des = {key: value for key, value in sorted(dt.items0. key=lambda item: item[1], reverse=True)}

print(asc)
print(des)
```

Output:

e) Write a Python script to check whether a given key already exists or not in a dictionary.

Program:

```
week = {'Mon':1,’Tue’:2,'Wed’:3,'Thu':4, 'Fri’:5, ’Sat':6, 'Sun':7}
print("Week Dictionary: ", week)
check_key = input("Enter a key to be verified in Dictionary: ”)
if check_key in week:
  print(check_key,” is Present.”)
else:
  print(check_key, " is not Present.")
```

Output:

f) Write a Python script to concatenate following dictionaries to create a new one:

dicl={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Program:

```
dicl={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4 = {}
for d in (dicl, dic2, dic3):
   dic4.update(d)
print(dic4)
```

Output:

g) Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are square of keys.

Program:

```
d=dict()
for x in range(1,16):
   d[x]=x**2
print(d)
```

Output:

h) Write a Python program to map two lists into a dictionary.

Program:

```
keys = ['red', 'green’, 'blue’]
values = ['#FF0000’,'#008000', '#0000FF']
color_dictionary = dict(zip(keys, values))
print(color_dictionary)
```

Output:

Module 5: Exercise Programs on Functions and Recursion

a) Define a function max„of„three() that takes three numbers as arguments and returns the largest of them.

Program:

```
n = [int(input("Enter Number %d : " %(i+1))) for i in range(3)]
def max_of_three(a,b,c):
   if(a>b  and a>c):
      return  a
   elif(b>c):
      return  b
   else:
      return  c
 print("Maximum Number is: ”, max_of_three(n[0], n[1], n[2]))
```

Output:

b) Write a program which **makes use of function** to display **all suchnumbers** which **are divisible by** 7 but **are** not **a multiple of** 5, **between** given **range** X and Y.

Program:

```
def numbers(a,b):
    x = [i for i in range(a,b+1) if(i%7 == 0 and i%5 != 0)]
    print("Numbers divisible by 7, but not by 5 between %d and %d are:" %(a,b))
    print(x)

start_range = int(input("Enter the Starting Range: ”))

end_range = int(input(”Enter the Ending Range: "))

numbers(start_range, end_range)
```

Output:

c) Define functions to find mean, median, mode for the given numbers ina list.

Program:

```
def mean(n, a):
    return sum(a)/n

def median(n,a):
    a.sort()
    if(n%2 == 0):
        return (a[(n-1)//2] + a[n//2])/2
    else:
        return a[n//2]

def mode(a):
    c = [a.count(i) for i in a]
    b = [i for i in range(1en(c)) if(c[i] == max(c))]
    return b[0]

n = int(input(”How many elements you want to enter into List? ”))
a = [int(input("Enter Element %d: " %(i+1))) for i in range(n)]
print("Mean = %.2f' %(mean(n,a)))
print("Median = %.2f' %(median(n,a)))
print("Mode = %d" %(a[(mode(a))]))
```

Output:

d) Define a function which generates Fibonacci series up to n numbers.

Program:

```
def fib(n):
   a = 0
   b =1
  if n == 1:
     print(a)
  else:
     print(a)
     print(b)
     for i in range(2,n):
        c = a + b
        a = b
        b = c
        print(c)
num = int(input(”Number that you want to Print Fibonacci Sequence: "))
fib(num)
```

Output:

e) Implement a python script for factorial of number by using recursion.

Program:

```
def recur_factoria1(n):
  if n == 1:
     return n
  else:
     return n*recur_factorial(n-1)

num = int(input(”Enter a number: "))

if num < 0:
  print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
  print("The factorial of 0 is 1")
else:
  print("The factorial of',num,"is",recur_factorial(num))
```

Output:

f) Implement a python script to find GCD of given two numbers using recursion.

Program:

```
def gcd_fun (x, y):
   if (y == 0):
      return x
   else:
      return gcd_fun (y, x % y)
x =int (input ("Enter the first number: "))
y =int (input (”Enter the second number: "))
num = gcd_fun(x, y)
print("GCD of two number is: ”, num)
```

Output:

Module 6: Exercise Programs on Strings

a) Implement Python Script to perform various operations on string using string libraries.

Program:

```
name = "Ram Kumar"
name_1 = "Ram_Kumar"
print(name.split())
print(name_1.split('_'))
print(name.isalnum())
print(name.isdigit())
print(name_1.lower())
print(name_1.upper())
print(name.capitalize())
print(len(name))
print(name_1.title())
```

Output:

b) Implement Python Script to check given string is palindrome or not.

Program:

```
my_str = input("Enter any string: ")
my_str = my_str.casefold()
rev_str = reversed(my_str)

if list(my_str) == list(rev_str):
  print("The string is a palindrome.")
else:
  print("The string is not a palindrome.”)
```

Output:

c) Implement python script to accept line of text and find the number of characters, number of vowels and number of blank spaces in it.

Program:

```
strl = input(”Please Enter Your Own String : ")
vowels = 0
consonants = 0

for i in str 1:
   if(i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u'
      or i == 'A' or i == 'E' or i == 'I’ or i == ’0' or i == 'U'):
       vowels = vowels + 1
   else:
       consonants = consonants + 1

print("Total Number of Vowels in this String = ", vowels)
print("Total Number of Consonants in this String = ”, consonants)
```

Output:

d) Implement python script that takes a list of words and returns the length of the longest one.

Program:

```
def find_longest_word(words_list):
    word_len = []
    for n in words_list:
        word_len.append((1en(n), n))
    word_len.sort()
    return word_len[-1][0], word_len[-1][1]
result=find_longest_word([”PHP","Exercises", "Backend”])
print("\nLongest word: ",result[1])
print("Length of the longest word: ",result[0])
```

Output:

Module 7: Exercise Programs on Regular Expressions

a) **Write a Python script to check that a string contains only a certain set of characters (in this case a-z,** A-Z **and** 0-9).

Program:

```
import re
def is_allowed_specific_char(string):
   charRe =re.compile(r’[^a-zA-Z0-9.]')
   string = charRe.search(string)
   return not bool(string)

string = input("Enter a String to verify: ")

print(is_allowed_specific_char(string))
```

Output:

b) Write a Python script to check whether password is valid or not.

Conditions for a valid password are:

1. Should have at least onenumber.
2. Should have at least one uppercase and one lowercasecharacter.
3. Should have at least one special symbol.
4. Should be between 6 to 20 characters long.

Program:

```
import re
password = input("Enter a Password: ")
x = True
while x:
    if (len(password)<6 orlen(password)>12):
        break
    elif not re.search(”[a-z]", password):
        break
    elif not re.search(”[0-9]", password):
        break
    elif not re.search(”[A-Z]",password):
        break
    elif not re.search(”[$#@]", password):
        break
    elifre.search("\s",password):
        break
    else:
        print("You Password is Valid")
        x=False
        break
if x:
    print(”Your Password is INVALID")
```

Output:

Module 8: Exercise Programs on Matplotlib library

a) Write a Python Program to draw a line with suitable label in x axis, y axis and atitle

```
import matplotlib.pyplot as plt
# x axis values
x = [1,2,3]
# y axis values
y = [2,4,1]
# Plot lines and/or markers to the Axes.
plt.plot(x, y)
# Set the x axis label of the current axis.
plt.xlabel('x - axis')
# Set the y axis label of the current axis.
plt.ylabel('y - axis')
# Set a title
plt.title('Sample graph!')
# Display a figure.
plt.show()
```

OUTPUT:

b) Write a python program to plot two or more lines with legends, different widths and colors.

```
 import matplotlib.pyplot as plt
# line 1 points
x1 = [10,20,30]
y1 = [20,40,10]
# line 2 points
x2 = [10,20,30]
y2 = [40,10,30]
# Set the x axis label of the current axis.
plt.xlabel('x - axis')
# Set the y axis label of the current axis.
plt.ylabel('y - axis')
# Set a title
plt.title('Two or more lines with different widths and colors with suitable legends')
# Display the figure.
plt.plot(x1,y1, color='blue', linewidth = 3, label = 'line1-width-3')
plt.plot(x2,y2, color='red', linewidth = 5,  label = 'line2-width-5')
# show a legend on the plot
plt.legend()
plt.show()
```

OUTPUT:

c) Write a Python Program to create Multiple plots.

```
  import matplotlib.pyplot as plt
fig = plt.figure()
fig.subplots_adjust(bottom=0.020, left=0.020, top = 0.900, right=0.800)

plt.subplot(2, 1, 1)
plt.xticks(()), plt.yticks(())

plt.subplot(2, 3, 4)
plt.xticks(())
plt.yticks(())

plt.subplot(2, 3, 5)
plt.xticks(())
plt.yticks(())

plt.subplot(2, 3, 6)
plt.xticks(())
plt.yticks(())

    plt.show()
```

Output:

d) Write a python program to display a bar chart using different color for each bar

```
import matplotlib.pyplot as plt

#data
x = ['cse', 'it', 'eee', 'ece', 'mech']
h = [10, 8, 7, 9, 5]
c = ['red', 'yellow', 'black', 'blue', 'orange']

#bar plot
plt.bar(x, height = h, color = c)
plt.show()
```

Output:

e) Write a Python programming to create a pie chart with a title.

Program:

```
import matplotlib.pyplot as plt

marks = [79, 82, 75, 96]

subjects = ['C', 'JAVA’, ’Python', 'WT']

plt.pie(marks, 1abels=subjects, autopct ='%1.1f%%')

p1t.title("Student Marks")

plt.legend()

plt.show()
```

Output:

F) Write a python program to draw a scatterplot with empty circles taking a random distribution in X and Y plotted against each other

```
import matplotlib.pyplot as plt

import numpy as np

x = np.random.randn(50)

y = np.random.randn(50)

plt.scatter(x, y, s=70, facecolors='none', edgecolors='g')

plt.xlabel("X")

plt.ylabel("Y")

plt.show()
```

Output: