**1.**Create a list of 10 elements of four different data types like int, string, complex and float.

a=[2,4,6,1.234,4.5,6.5,'Ankur','Pathania',3.14j,1.66j]

print(type(a))

**2.** Create a list of size 5 and execute the slicing structure.

a = [10, 30, 20, 40, 60, 50]

print(a[1:5])

**3.** Write a program to get the sum and multiply of all the items in a given list.

def multiply\_list(items):

    a = 1

    for x in items:

        a \*= x

    return a

print(multiply\_list([3,2,1]))

def sum\_list(items):

    a = 0

    for x in items:

        a += x

    return a

print(sum\_list([3,2,1,10]))

**4.** Find the largest and smallest number from a given list.

lst = []

num = int(input('How many numbers: '))

for n in range(num):

    numbers = int(input('Enter number '))

    lst.append(numbers)

print("Maximum element in the list is :", max(lst), "\nMinimum element in the list is :", min(lst))

**5.** Create a new list which contains the specified numbers after removing the even numbers from a predefined list.

num = [1, 2, 3, 4, 5, 6, 7]

num = [x for x in num if x%2!=0]

print(num)

**6.** Create a list of elements such that it contains the squares of the first and last 5 elements between 1 and30 (both included).

def printValues():

    l = list()

    for i in range(1,31):

        l.append(i\*\*2)

    print(l[:5])

    print(l[-5:])

printValues()

**7.** Write a program to replace the last element in a list with another list.

**Sample input:** [1,3,5,7,9,10], [2,4,6,8]

**Expected output:** [1,3,5,7,9,2,4,6,8]

num1 = [1, 3, 5, 7, 9, 10]

num2 = [2, 4, 6, 8]

num1[-1:] = num2

print(num1)

**8.** Create a new dictionary by concatenating the following two dictionaries:

**Sample input:** a={1:10,2:20} b={3:30,4:40}

**Expected output:** {1:10,2:20,3:30,4:40}

a={1:10, 2:20}

b={3:30, 4:40}

c = {}

for d in (a, b): c.update(d)

print(c)

**9.** Create a dictionary that contain numbers in the form(x:x\*x) where x takes all the values between 1 and n(both 1 and n included).

**Sample input:** n=5

**Expected output:** {1:1, 2:4, 3:9, 4:16, 5:25}

n=int(input("Input a number "))

d = dict()

for x in range(1,n+1):

    d[x]=x\*x

print(d)

**10.** Write a program which accepts a sequence of comma-separated numbers from console and generates a list and a tuple which contains every number in the form of string.

**Sample input:** 34,67,55,33,12,98

**Expected output:** [‘34’,’67’,’55’,’33’,’12’,’98’] (‘34’,’67’,’55’,’33’,’12’,’98’)

values = input("Enter the comma seprated numbers: ")

list = values.split(",")

tuple = tuple(list)

print('List : ',list)

print('Tuple : ',tuple)

**TASK FOUR**

**TRADITIONAL FUNCTIONS,ANONYMOUS FUNCTIONS &**

**HIGHER ORDER FUNCTIONS**

**1.** Write a program to reverse a string.

**Sample input:** “1234abcd”

**Expected output:** “dcba4321”

def reverse\_string(str1):

    rstr1 = ''

    index = len(str1)

    while index > 0:

        rstr1 += str1[ index - 1 ]

        index = index - 1

    return rstr1

print(reverse\_string('1234abcd'))

**2.** Write a function that accepts a string and prints the number of uppercase letters and lowercase letters.

**Sample input:** “abcSdefPghijQkl”

**Expected Output:** No. of Uppercase characters : 3 No. of Lower case Characters : 12

def string\_test(s):

    d={"UPPER\_CASE":0, "LOWER\_CASE":0}

    for c in s:

        if c.isupper():

           d["UPPER\_CASE"]+=1

        elif c.islower():

           d["LOWER\_CASE"]+=1

        else:

           pass

    print ("Original String : ", s)

    print ("No. of Upper case characters : ", d["UPPER\_CASE"])

    print ("No. of Lower case Characters : ", d["LOWER\_CASE"])

string\_test('abcSdefPghijQkl')

**3.** Create a function that takes a list and returns a new list with unique elements of the first list.

def unique\_list(l):

  x = []

  for a in l:

    if a not in x:

      x.append(a)

  return x

print(unique\_list([1,1,2,2,3,3,3,3,4,4,5,5]))

**4.** Write a program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

items=[n for n in input("Enter the value: ").split('-')]

items.sort()

print('-'.join(items))

**5.** Write a program that accepts a sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

**Sample input:** Hello world Practice makes man perfect

**Expected output:** HELLO WORLD PRACTICE MAKES MAN PERFECT

lines = []

while True:

    l = input("Enter the value(all lower-case)")

    if l:

        lines.append(l.upper())

    else:

        break;

for l in lines:

    print(l)

**6.** Define a function that can receive two integral numbers in string form and compute their sum and print it in the console.

def calculate\_sum (a,b):

    s = int(a) + int(b)

    return s

num1 = "10"

num2 = "20"

sum = calculate\_sum (num1, num2)

print ("Sum = ", sum)

**7.** Define a function that can accept two strings as input and print the string with the maximum length in the console. If two strings have the same length, then the function should print both the strings line by line.

def length\_of\_string(str1, str2):

    if (len(str1) == len(str2)):

        print(str1)

        #print("\n")

        print(str2)

    elif (len(str1) < len(str2)):

        print(str2)

    else:

        print(str1)

stri1 = input(str("enter First String: "))

stri2 = input(str("enter Second String: "))

print("\n")

length\_of\_string(stri1, stri2)

**8.** Define a function which can generate and print a tuple where the values are square of numbers between 1 and 20 (both 1 and 20 included).

def printValues():

    l = list()

    for i in range(1,21):

        l.append(i\*\*2)

    print(l)

printValues()

**9.** Write a function called showNumbers that takes a parameter called limit. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers.

**Sample input:** show Numbers(3) (where limit=3)

**Expected output:**

0 EVEN

1 ODD

2 EVEN

3 ODD

def shownumber(limit):

    for i in range(0, limit):

        if i==0:

            print(i,end=" ")

            print("EVEN")

        elif i%2==0:

            print(i,end=" ")

            print("EVEN")

        else:

            print(i,end=" ")

        print("ODD")

    print(shownumber(4))

**10.** Write a program which uses filter() to make a list whose elements are even numbers between 1 and 20 (both included)

nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18,19,20]

print("Original list of integers:")

print(nums)

print("\nEven numbers from the said list:")

even\_nums = list(filter(lambda x: x%2 == 0, nums))

print(even\_nums)

**11.** Write a program which uses map() and filter() to make a list whose elements are squares of even numbers in [1,2,3,4,5,6,7,8,9,10].

Hints: Use filter() to filter even elements of the given listUse map() to generate a list of squares of the numbers in the filtered list. Use lambda() to define anonymous functions.

li = [1,2,3,4,5,6,7,8,9,10]

eve\_num = map(lambda x: x\*\*2, filter(lambda   x: x%2==0, li))

print(eve\_num)

**12.** Write a function to compute 5/0 and use try/except to catch the exceptions.

def divide():

    return 5/0

try:

    divide()

except ZeroDivisionError as ze:

    print("Dividing a number by ZERO!!")

except:

    print("Any other exception")

**13.** Flatten the list [1,2,3,4,5,6,7] into 1234567 using reduce().

import itertools

List\_2D = [1,2,3,4,5,6,7,8,9]

List\_flat = int("".join(map(str, [1,2,3,4,5,6,7,8,9])))

print("Original List:",List\_2D)

print("Flattened List:",List\_flat)

**14.** Write a program in Python to find the values which are not divisible by 3 but are a multiple of 7. Make sure to use only higher order functions.

**15**. Write a program in Python to multiply the elements of a list by itself using a traditional function and pass the function to map() to complete the operation.

def square(x):

    return x \* x

nums = [1, 2, 3, 4, 5]

nums\_squared = map(square, nums)

for num in nums\_squared:

    print(num)

**16.** What is the output of the following codes:

**(i)** def foo():

try:

return 1

finally:

return 2

k = foo()

print(k)

**O/P: 2**

(ii) def a():

try:

f(x, 4)

finally:

print('after f')

print('after f?')

a()

**O/P: Blank**