**AP LAB – SESSION 2**

**LAB 3:**

QUESTION – 1:

#QUESTION 1

# define a function to multiply all elements in the list

l = [int(i) for i in input("Enter a list: ").split()]

def mult (a) :

“Finding the product of elements of the list”

product = 1

for ele in a :

product \*= ele

return product

print("Multiplication of elements in given list = ", mult(l))



QUESTION 2:

#QUESTION 2

# define a function to multiply all elements in the list

l = [int(i) for i in input("Enter a list: ").split()]

def unique\_ele (a) :

"return all unique elementsin the given list"

return list(set(a))

print("List of unique elements in given list: ", unique\_ele(l))



**LAB 4:**

QUESTION – 1:

#QUESTION 1

#Using built-in modules to find sin, sqrt and log values of given number

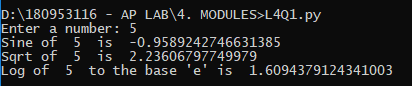
import math

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

print("Sine of ",n," is ", math.sin(n))

print("Sqrt of ",n," is ", math.sqrt(n))

print("Log of ",n," to the base 'e' is ", math.log(n))



QUESTION – 2:

#QUESTION 2

#Using built-in modules to find sin, sqrt and log values of given complex number

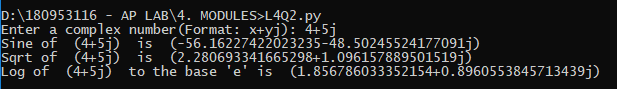
import cmath

n = complex(input("Enter a complex number(Format: x+yj): "))

print("Sine of ",n," is ", cmath.sin(n))

print("Sqrt of ",n," is ", cmath.sqrt(n))

print("Log of ",n," to the base 'e' is ", cmath.log(n))



QUESTION – 3:

#QUESTION 3

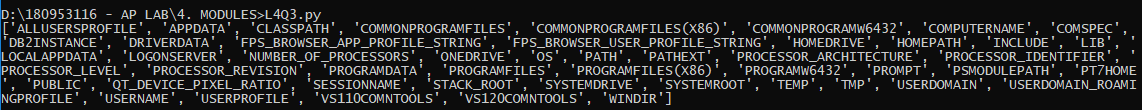
#List all environment variables

import os

var = os.environ

l\_var = list(var)

print(l\_var)



**LAB 5:**

QUESTION – 1:

#QUESTION 1

#Store employee details in tuples and search based on ID

class Employee :

def \_\_init\_\_(self, emp) :

self.t = tuple(emp)

def search\_id(self, iden) :

if self.t[0] == iden :

return "Details are: {}".format(self.t)

else :

return False

n = int(input("Enter number of employees: "))

empl = list()

for i in range(n):

print("Enter details of " + str(i+1) + "th employee: (id, name, salary, department)")

emp = input().split()

empl.append(Employee(emp))

iden = input("Enter I'd of the employee to be searched: ")

for e in empl :

val = e.search\_id(iden)

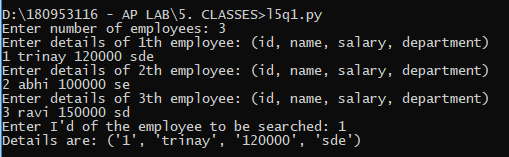
if val != False :

print(val)

break

else :

print("No employee found with I'd = ",iden)



QUESTION – 3:

# QUESTION 3

# Get all unique subsets using class

class subset\_listing :

def \_\_init\_\_(self, num\_list) :

self.num\_list = num\_list

def print\_subsets(self) :

subset\_list = list()

append\_zeroes = "0"

len\_list = len(self.num\_list)

for i in range(2\*\*len\_list) :

bin\_i = str(bin(i))[2:]

bin\_i = append\_zeroes\*(len\_list - len(bin\_i)) + bin\_i

subset = {self.num\_list[j] for j, bit in enumerate(bin\_i) if bit == '1'}

subset\_list.append(subset)

for ele in subset\_list :

print (ele)

l = [int(i) for i in input("Enter a list of integers: ").split()]

l = list(set(l))

obj = subset\_listing(l)

obj.print\_subsets()

