SESSION 2 – 9/12/2020

LAB 3

Q1. SOURCE CODE

def multiply(l1):

res = 1

for num in l1:

res = res \* num

return res

x = input("Enter the elements of the list:")

list1 = x.split()

for i in range(len(list1)):

list1[i] = int(list1[i])

print("Result of multiplication of each element of the list = ", multiply(list1))

SAMPLE OUTPUT

Enter the elements of the list:3 -2 4 10 2

Result of multiplication of each element of the list = -480

Q2. SOURCE CODE

x = input("Enter the list elements:")

l1 = x.split()

#str to int

for i in range(len(l1)):

l1[i] = int(l1[i])

a = dict()

for i in l1:

if i not in a:

a[i] = 1

else:

a[i] += 1

newlist = []

for keys in a:

if a[keys] == 1:

newlist.append(keys)

print("The unique elements are ",newlist)

SAMPLE OUTPUT

Enter the list elements:4 5 2 4 7 5 8 54 2

The unique elements are [54, 7, 8]

LAB 4

Q1. SOURCE CODE(module part of the code)

#!/usr/bin/python3

import math

def sine(num):

return math.sin(num)

def logarithm(num2):

return math.log(num2)

def squareroot(num3):

return math.sqrt(num3)

SOURCE CODE(main code)

import mod1

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

print("sin("+str(x)+") = "+str(mod1.sine(x)))

print("square\_root("+str(x)+") = "+str(mod1.squareroot(x)))

print("log("+str(x)+") = "+str(mod1.logarithm(x)))

SAMPLE OUTPUT

Enter any number:81

sin(81) = -0.6298879942744539

square\_root(81) = 9.0

log(81) = 4.394449154672439

Q2. SOURCE CODE(module part)

#!/usr/bin/python3

import cmath

def sine(num):

return cmath.sin(num)

def logarithm(num2):

return cmath.log(num2)

def squareroot(num3):

return cmath.sqrt(num3)

SOURCE CODE(main part)

import mod2

cnum = complex(input("Enter any complex number:"))

print("sin"+str(cnum)+" = "+str(mod2.sine(cnum)))

print("log"+str(cnum)+" = "+str(mod2.logarithm(cnum)))

print("square\_root"+str(cnum)+" = "+str(mod2.squareroot(cnum)))

SAMPLE OUTPUT

Enter any complex number: 49-49j

sin(49-49j) = (-9.095684266244439e+20-2.8666707901350478e+20j)

log(49-49j) = (4.2383938883906-0.7853981633974483j)

square\_root(49-49j) = (7.6907887942746695-3.1856290239355913j)

Q3. SOURCE CODE

import os

#using the os module to print all the environment variables

print("Printing all the environment variables:")

for k,v in os.environ.items():

print("{} = {}".format(k,v))

SAMPLE OUTPUT( all the environment variables )

Printing all the environment variables:

WINDIR = C:\Windows

PROGRAMFILES(X86) = C:\Program Files (x86)

USERNAME = student

FP\_NO\_HOST\_CHECK = NO

COMPUTERNAME = MIT-ICT-LAB8-26

TCL\_LIBRARY = C:\Python34\tcl\tcl8.6

ALLUSERSPROFILE = C:\ProgramData

PROCESSOR\_REVISION = 3c03

PT5HOME = C:\Program Files (x86)\Cisco Packet Tracer 6.1sv

USERDOMAIN = MIT-ICT-LAB8-26

PROGRAMDATA = C:\ProgramData

HOMEPATH = \Users\student

COMMONPROGRAMW6432 = C:\Program Files\Common Files

NVTOOLSEXT\_PATH = C:\Program Files\NVIDIA Corporation\NvToolsExt\

APPDATA = C:\Users\student\AppData\Roaming

NVCUDASAMPLES\_ROOT = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v5.5

PROCESSOR\_ARCHITECTURE = AMD64

PROCESSOR\_IDENTIFIER = Intel64 Family 6 Model 60 Stepping 3, GenuineIntel

NUTSUFFIX = 1

TEMP = C:\Users\student\AppData\Local\Temp

TK\_LIBRARY = C:\Python34\tcl\tk8.6

PROGRAMW6432 = C:\Program Files

CUDA\_PATH\_V5\_5 = C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.5

NUT\_SUFFIXED\_SEARCHING = 1

HOMEDRIVE = C:

SESSIONNAME = Console

PATH = C:\Perl\site\bin;C:\Perl\bin;C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.5\bin;C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.5\libnvvp;C:\app\client\student\product\12.1.0\client\_1;C:\app\client\student\product\12.1.0\client\_1\bin;C:\app\student\product\11.1.0\client\_1\bin;C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\;C:\Program Files\MySQL\MySQL Server 5.5\bin;C:\Program Files (x86)\Rational\common;C:\Program Files (x86)\Rational\ClearCase\bin;C:\MinGW\bin;C:\msys\1.0\bin;C:\Program Files (x86)\MiKTeX 2.9\miktex\bin\;C:\Program Files (x86)\Windows Live\Shared;C:\Program Files (x86)\Flash Magic;C:\Program Files\MATLAB\R2019a\runtime\win64;C:\Program Files\MATLAB\R2019a\bin;C:\Program Files (x86)\Windows Kits\8.1\Windows Performance Toolkit\;C:\Program Files\SDCC\bin;C:\Users\student\AppData\Local\Programs\Microsoft VS Code\bin

TIX\_LIBRARY = C:\Python34\tcl\tix8.4.3

TISDIR = C:\Program Files (x86)\Rational\common

USERDOMAIN\_ROAMINGPROFILE = MIT-ICT-LAB8-26

COMSPEC = C:\Windows\system32\cmd.exe

CLASSPATH = .

OS = Windows\_NT

SYSTEMDRIVE = C:

PATHEXT = .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY

TMPDIR = C:\Users\student\AppData\Local\Temp

PROCESSOR\_LEVEL = 6

NVCUDASAMPLES5\_5\_ROOT = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v5.5

PSMODULEPATH = C:\Windows\system32\WindowsPowerShell\v1.0\Modules\

CUDA\_PATH = C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.5

COMMONPROGRAMFILES(X86) = C:\Program Files (x86)\Common Files

VS120COMNTOOLS = C:\Program Files (x86)\Microsoft Visual Studio 12.0\Common7\Tools\

USERPROFILE = C:\Users\student

SYSTEMROOT = C:\Windows

HOME = C:\Users\student

IBMLDAP\_ALTHOME = C:\Program Files (x86)\Rational\common\codeset

PT6HOME = C:\Program Files (x86)\Cisco Packet Tracer 6.1sv

PROGRAMFILES = C:\Program Files

TMP = C:\Users\student\AppData\Local\Temp

RATL\_RTHOME = C:\Program Files (x86)\Rational\Rational Test

PUBLIC = C:\Users\Public

VS110COMNTOOLS = C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\Tools\

LOCALAPPDATA = C:\Users\student\AppData\Local

VBOX\_INSTALL\_PATH = C:\Program Files\Oracle\VirtualBox\

NUMBER\_OF\_PROCESSORS = 8

LOGONSERVER = \\MIT-ICT-LAB8-26

COMMONPROGRAMFILES = C:\Program Files\Common Files

>>>

LAB 5

Q1. SOURCE CODE

class employee:

def \_\_init\_\_(self, Id, name, sal, dept):

self.Id = Id

self.name = name

self.sal = sal

self.dept = dept

def create\_tuple(self):

return (self.Id, self.name, self.sal, self.dept)

def search\_emp(self, search):

flag = 0

for record in emp\_details:

if search in record[0]:

print("Entry found:")

print("ID: {}".format(record[0]))

print("Name = {}".format(record[1]))

print("Salary: {}".format(record[2]))

print("Department: {}".format(record[3]))

flag = 1

if flag == 0:

print("No employee found with ID = {}".format(search))

emp\_details = []

n = int(input("How many employees:"))

for i in range(n):

Id = input("ID:")

name = input("Name:")

sal = input("Salary:")

dept = input("Department:")

emp = employee(Id, name, sal, dept)

emp\_details.append(emp.create\_tuple())

print("All details entered successfully")

search = input("Enter the employee ID to search:")

emp.search\_emp(search)

print("\nAll the employees:")

for record in emp\_details:

print(record)

SAMPLE OUTPUT

How many employees:3

ID:6758

Name:YASH

Salary:45678

Department:MARKETING

ID:4454

Name:ARYAN

Salary:56789

Department:RESEARCH

ID:4543

Name:HARSH

Salary:50000

Department:PLANNING

All details entered successfully

Enter the employee ID to search:4454

Entry found:

ID: 4454

Name = ARYAN

Salary: 56789

Department: RESEARCH

All the employees:

('6758', 'YASH', '45678', 'MARKETING')

('4454', 'ARYAN', '56789', 'RESEARCH')

('4543', 'HARSH', '50000', 'PLANNING')

Q3. SOURCE CODE

class unique\_num:

num = []

all\_subs = []

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

self.num = num

def powerset(self,seq):

#in all cases, one entry will be null set and the other entire set

if len(seq)<=1:

yield seq

yield []

else:

for item in self.powerset(seq[1:]):

yield [seq[0]]+item

yield item

n = input("Enter unique numbers:")

n = n.split()

#converting to string to integer

for i in range(len(n)):

n[i] = int(n[i])

num\_class = unique\_num(n) #class

x = [l for l in num\_class.powerset(n)]

print(x)

SAMPLE OUTPUT

Enter unique numbers:1 2

[[1, 2], [2], [1], []]

Enter unique numbers:5 6 7 8

[[5, 6, 7, 8], [6, 7, 8], [5, 7, 8], [7, 8], [5, 6, 8], [6, 8], [5, 8], [8], [5, 6, 7], [6, 7], [5, 7], [7], [5, 6], [6], [5], []]