In [ ]:

```
# LAB 3
```

In [ ]:

```
# QUESTION 1 - Multiply all numbers in a list
```

In [7]:

```
def mulList(numList):
    if(len(numList)==0):
        return None
    product = 1
    for num in numList:
        product*=num
    return product

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')

n = int(input('Enter size of list:'))
numList = []
for i in range(n):
    val = int(input('Enter value:'))
    numList.append(val)

result = mulList(numList)
if(result==None):
    print('Invalid input!')
else:
    print('The product of the numbers in the list is:', result)
```

```
OUTPUT
------
Nilesh Jain - 180953226
Enter size of list:5
Enter value:2
Enter value:5
Enter value:-2
Enter value:3
Enter value:6
The product of the numbers in the list is: -360
```

In [ ]:

```
# QUESTION 2 - Return new list with unique elements of input list
```

In [13]:

```
def uniqueList(numList):
    if(len(numList)==0):
        return None
    unique = set(numList)
    return list(unique)

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')

n = int(input('Enter size of list:'))
numList = []
for i in range(n):
    val = int(input('Enter value:'))
    numList.append(val)

result = uniqueList(numList)
if(result==None):
    print('Invalid input!')
else:
    print('The unique elements of the list are:', end = " ")
    for i in range(len(result)):
        print(result[i], end=" ")
```

```
OUTPUT
------
Nilesh Jain - 180953226
Enter size of list:5
Enter value:1
Enter value:1
Enter value:2
Enter value:2
Enter value:3
The unique elements of the list are: 1 2 3
```

In [ ]:

```

```

In [ ]:

```
# LAB 4
```

In [ ]:

```
# QUESTION 1 - accept number, and print sine values, square values and log value
s
```

In [21]:

```
import math # math module deals with real numbers

def dispVal(n):
    print('Sine value is:', math.sin(n))
    print('Square root is:', math.sqrt(n))
    print('Log is:', math.log(n,10)) # second parameter is base

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')

n = int(input('Enter number:'))
dispVal(n)
```

```
OUTPUT
------
Nilesh Jain - 180953226
Enter number:10
Sine value is: -0.5440211108893698
Square root is: 3.1622776601683795
Log is: 1.0
```

In [ ]:

```
# QUESTION 2 - Repeat question 1, but with complex numbers as input
```

In [22]:

```
import cmath # cmath module deals with complex numbers
def dispVal(n):
    print('Sine value is:', cmath.sin(n))
    print('Square root is:', cmath.sqrt(n))
    print('Log is:', cmath.log(n,10)) # second parameter is base

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')

x = int(input('Enter Real Part:'))
y = int(input('Enter Complex Part:'))
n = complex(x,y)
print('Your complex number is:',n)
dispVal(n)
```

```
OUTPUT
------
Nilesh Jain - 180953226
Enter Real Part:3
Enter Complex Part:4
Your complex number is: (3+4j)
Sine value is: (3.853738037919377-27.016813258003932j)
Square root is: (2+1j)
Log is: (0.6989700043360187+0.4027191962733731j)
```

In [ ]:

```
# QUESTION 3 - list all environment variables
```

In [24]:

```
import os # module to interact with operating system

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226\n')
print(os.environ) # dictionary of all environment variables
```

```
OUTPUT
------
Nilesh Jain - 180953226

environ({'SHELL': '/bin/bash', 'SESSION_MANAGER': 'local/inspiron-75
70:@/tmp/.ICE-unix/6480,unix/inspiron-7570:/tmp/.ICE-unix/6480', 'QT
_ACCESSIBILITY': '1', 'NVM_RC_VERSION': '', 'COLORTERM': 'truecolo
r', 'XDG_CONFIG_DIRS': '/etc/xdg/xdg-ubuntu:/etc/xdg', 'XDG_MENU_PRE
FIX': 'gnome-', 'GNOME_DESKTOP_SESSION_ID': 'this-is-deprecated', 'L
ANGUAGE': 'en_IN:en', 'MANDATORY_PATH': '/usr/share/gconf/ubuntu.man
datory.path', 'GNOME_SHELL_SESSION_MODE': 'ubuntu', 'SSH_AUTH_SOCK':
'/run/user/1000/keyring/ssh', 'XMODIFIERS': '@im=ibus', 'DESKTOP_SES
SION': 'ubuntu', 'SSH_AGENT_PID': '6439', 'GTK_MODULES': 'gail:atk-b
ridge', 'PWD': '/home/njeyepatch/Manipal Study Material', 'LOGNAME':
'njeyepatch', 'XDG_SESSION_DESKTOP': 'ubuntu', 'QT_QPA_PLATFORMTHEM
E': 'qt5ct', 'XDG_SESSION_TYPE': 'x11', 'GPG_AGENT_INFO': '/run/use
r/1000/gnupg/S.gpg-agent:0:1', 'XAUTHORITY': '/run/user/1000/gdm/Xau
thority', 'GJS_DEBUG_TOPICS': 'JS ERROR;JS LOG', 'WINDOWPATH': '2',
'HOME': '/home/njeyepatch', 'USERNAME': 'njeyepatch', 'IM_CONFIG_PHA
SE': '1', 'LANG': 'en_IN', 'LS_COLORS': 'rs=0:di=01;34:ln=01;36:mh=0
0:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=
00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.
tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;3
1:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo
=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz
=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.
bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;3
1:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=
01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.
rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;3
1:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.
bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;3
5:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svg
z=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;3
5:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v
=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:
*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=0
1;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv
=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:
*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=0
0;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.og
a=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:', 'XDG_CURRENT_DESKTO
P': 'ubuntu:GNOME', 'VTE_VERSION': '6003', 'GNOME_TERMINAL_SCREEN':
'/org/gnome/Terminal/screen/4fd5a140_1ffb_48fb_b24a_3512cb27f739',
'INVOCATION_ID': '1d6c01ed4b894910a250ee6486e83647', 'MANAGERPID':
'6229', 'GJS_DEBUG_OUTPUT': 'stderr', 'NVM_DIR': '/home/njeyepatch/.
nvm', 'QT_DEVICE_PIXEL_RATIO': 'auto', 'LESSCLOSE': '/usr/bin/lesspi
pe %s %s', 'XDG_SESSION_CLASS': 'user', 'TERM': 'xterm-color', 'DEFA
ULTS_PATH': '/usr/share/gconf/ubuntu.default.path', 'LESSOPEN': '| /
usr/bin/lesspipe %s', 'USER': 'njeyepatch', 'GNOME_TERMINAL_SERVIC
E': ':1.507', 'DISPLAY': ':0', 'SHLVL': '1', 'NVM_CD_FLAGS': '', 'PT
7HOME': '/opt/pt', 'QT_IM_MODULE': 'ibus', 'XDG_RUNTIME_DIR': '/run/
user/1000', 'JOURNAL_STREAM': '8:51099', 'XDG_DATA_DIRS': '/usr/shar
e/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop', 'PAT
H': '/home/njeyepatch/.local/bin:/usr/local/sbin:/usr/local/bin:/us
r/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/us
r/local/go/bin', 'GDMSESSION': 'ubuntu', 'DBUS_SESSION_BUS_ADDRESS':
'unix:path=/run/user/1000/bus', 'OLDPWD': '/home/njeyepatch/Manipal
Study Material/advanced-programming-lab', '_': '/home/njeyepatch/.lo
cal/bin/jupyter-notebook', 'JPY_PARENT_PID': '1356050', 'CLICOLOR':
```

```
'1', 'PAGER': 'cat', 'GIT_PAGER': 'cat', 'MPLBACKEND': 'module://ipy
kernel.pylab.backend_inline'})
```

In [ ]:

```
```

In [ ]:

```
# LAB 5
```

In [ ]:

```
# QUESTION 1 - read details of n employees
```

In [35]:

```
class employee():
    employees = [] # list of all employees
    def __init__(self,ID,name,salary,department):
        self.ID = ID
        self.name = name
        self.salary = salary
        self.department = department
        employee.employees.append((ID,name,salary,department))

    @staticmethod # the employees array is a class variable and hence we declare
this as a static method
    def search(ID):
        found = False
        position = -1
        for i in range(len(employee.employees)):
            if(employee.employees[i][0]==ID):
                found = True
                position = i
        if(found==False):
            print('Employee Not Found!')
        else:
            employee.display(position)

    @staticmethod
    def display(position):
        print('ID:',employee.employees[position][0])
        print('Name:',employee.employees[position][1])
        print('Salary:',employee.employees[position][2])
        print('Department:',employee.employees[position][3])

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print("\n")

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

for i in range(n):
    ID = int(input('Enter id:'))
    name = input('Enter name:')
    salary = float(input('Enter salary:'))
    department = input('Enter department:')
    print("\n")
    employee(ID,name,salary,department)

searchID = int(input('Enter ID to be searched:'))
print("\n")
employee.search(searchID)
```

```
OUTPUT
------
Nilesh Jain - 180953226


Enter number of employees:3


Enter id:1
Enter name:Nilesh
Enter salary:50000
Enter department:IT


Enter id:2
Enter name:Anoop
Enter salary:30000
Enter department:HR


Enter id:3
Enter name:Ram
Enter salary:40000
Enter department:Marketing


Enter ID to be searched:2


ID: 2
Name: Anoop
Salary: 30000.0
Department: HR
```

In [ ]:

```
# QUESTION 2 - create a class vehicle and derived class passenger, and take inpu
t from user and display the same
```

In [36]:

```
class vehicle():
     def __init__(self): # properties of vehicle class
            self.vid = None
            self.name = None
            self.mfd = None

class passenger(vehicle):
    def __init__(self): # properties of passenger class
        self.passengers = None

    def getInput(self):
            self.vid = input('Enter vehicle ID:')
            self.name = input("Enter owner's name:")
            self.mfd = input('Enter name of manufacturer:')
            self.passengers = input('Enter number of passengers:')

    def display(self):
        print('Vehicle ID:', self.vid)
        print('Name of Owner:',self.name)
        print('Name of Manufacturer:',self.mfd)
        print('Number of Passengers:',self.passengers)
        print("\n")
print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print("\n")

n = int(input('Enter number of vehicles:'))
vehicles = []

for i in range(n):
    vehicles.append(passenger())
    vehicles[i].getInput()

print('\nDisplaying all vehicles data:')
print("\n")

for i in range(n):
    vehicles[i].display()
```

```
OUTPUT
------
Nilesh Jain - 180953226


Enter number of vehicles:3
Enter vehicle ID:TN104534
Enter owner's name:Nilesh
Enter name of manufacturer:Maruti Suzuki
Enter number of passengers:5
Enter vehicle ID:KL204567
Enter owner's name:Anoop
Enter name of manufacturer:Hyundai
Enter number of passengers:5
Enter vehicle ID:PY304466
Enter owner's name:Ram
Enter name of manufacturer:Honda
Enter number of passengers:7

Displaying all vehicles data:


Vehicle ID: TN104534
Name of Owner: Nilesh
Name of Manufacturer: Maruti Suzuki
Number of Passengers: 5


Vehicle ID: KL204567
Name of Owner: Anoop
Name of Manufacturer: Hyundai
Number of Passengers: 5


Vehicle ID: PY304466
Name of Owner: Ram
Name of Manufacturer: Honda
Number of Passengers: 7
```

In [ ]:

```
# QUESTION 3 - list all unique subsets
```

In [37]:

```
class subsets:
    def __init__(self, unique):
        self.subsets = [[]] # initial empty subset list
        self.unique = unique

    def subsetCreate(self):
         for i in range(len(self.unique)):
            initial = self.subsets[:] # all subsets before current
            new = self.unique[i]
            for j in range(len(self.subsets)):
                self.subsets[j] = self.subsets[j] + [new] # adding all elements
 possible in a subset
            self.subsets = initial + self.subsets # appending possible subsets t
o list

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print("\n")

print("Enter list of unique numbers:")
unique = list(map(int, input().split()))

subset = subsets(unique)
subset.subsetCreate()
print("Subsets are : ", subset.subsets)
```

```
OUTPUT
------
Nilesh Jain - 180953226


Enter list of unique numbers:
4 5 6
Subsets are :  [[], [4], [5], [4, 5], [6], [4, 6], [5, 6], [4, 5,
6]]
```