# PRACTICAL 1

**1[a]**

**AIM : Take an integer value (N) from the user and write a program to print even numbers between 0 and N.**

**INPUT :**

```
b = int(input("ENTER THE RANGE : "))
for i in range(0 , b+1):
    if i % 2 == 0 :
        print(i , end = " ")
        print("")
```

**OUTPUT :**

```
ENTER THE RANGE : 8
0
2
4
6
8
```

**1[b]**

**AIM : Write a program to print multiplication table of a given number using while loop.**

**INPUT :**

```
a = int(input("Enter the table :"))
print(" MULTIPLICATION TABLE OF : ",a )
i=1
while(i<=10):
  print(a ,"*",i ,"=",i*a)
  i=i+1
```

**OUTPUT :**

```
Enter the table : 13
MULTIPLICATION TABLE OF :  13
13 * 1 = 13
13 * 2 = 26
13 * 3 = 39
13 * 4 = 52
13 * 5 = 65
13 * 6 = 78
13 * 7 = 91
13 * 8 = 104
13 * 9 = 117
13 * 10 = 130
```

**1[c]**

**AIM : Ask the user to enter student name and marks in four subjects. Write a program to calculate percentage and evaluate grade of the student using the following criteria. The output should display "Hello < student_name >. you have achieved grade <grade>",**

**Grade: A if percentage 2 98**

**Grade: 8 if 90 percentage 2 75 Grade: if 75 percentage 2 50**

**Grade: D if 50 percentage 2 33**

**Grade: E if 33 percentage**

**INPUT :**

```
name = input("STUDENT NAME : ")

sub1 = int(input("MARKS OF SUB1 : "))
sub2 = int(input("MARKS OF SUB2 : "))
sub3 = int(input("MARKS OF SUB3 : "))
sub4 = int(input("MARKS OF SUB4 : "))

Marks = (sub1+sub2+sub3+sub4)
per = (Marks*100/400)

print("MARKS OBTAINED :" , Marks )
print("TOTAL MARKS : 400")
print("PERCENTAGE OBTAINED :", per )

if per >= 90:
   print("HELLO",name,".","YOU OBTAINED A GRADE")
```

```
elif per >= 75 and per < 90 :
   print("HELLO",name,".","YOU OBTAINED B GRADE")

elif per >= 50 and per < 75 :
   print("HELLO",name,".","YOU OBTAINED C GRADE")

elif per >= 33 and per < 50 :
   print("HELLO",name,".","YOU OBTAINED D9 GRADE")

elif per < 33  :
   print("HELLO",name,".","YOU OBTAINED E GRADE")
```

**OUTPUT :**

```
  STUDENT NAME : TOM
  MARKS OF SUB1 : 88
  MARKS OF SUB2 : 98
  MARKS OF SUB3 : 78
  MARKS OF SUB4 : 89
  MARKS OBTAINED : 353
  TOTAL MARKS : 400
  PERCENTAGE OBTAINED : 88.25
  HELLO TOM  . YOU OBTAINED B GRADE
```

**1[d]**

**AIM : Write a program to read text and calculate number of lowercase letters, uppercase letters, digits, alphabets, special characters, and words..**

**INPUT :**

```
def Count(str):
    upper, lower, number, special = 0, 0, 0, 0
    for i in range(len(str)):
        if str[i].isupper():
            upper += 1
        elif str[i].islower():
            lower += 1
        elif str[i].isdigit():
            number += 1
        else:
            special += 1
    print('Upper case letters:', upper)
    print('Lower case letters:', lower)
    print('Number:', number)
```

```
    print('Special characters:', special)

# Driver Code
str = "#pyt#0n&PYTHON$9854"
Count(str)
```

**OUTPUT :**

```
Upper case letters: 6
Lower case letters: 4
Number: 5
Special characters: 4
```

**1[e]**

**AIM : Ask the user to enter number(s) as many as he/she wants. Write a program to store these numbers in a list and find out the largest number.**

**INPUT :**

```
list = []
num = int(input("ENTER THE NUMBER OF ELEMENTS YOU WANT IN THE LIST : "))

for i in range(1,num+1):
  element = int(input("ENTER THE VALUE OF ELEMENTS : "))
  list.append(element)
print("LARGEST ELEMENT IS : " , max(list))
```

**OUTPUT :**

```
ENTER THE NUMBER OF ELEMENTS YOU WANT IN THE LIST : 7
ENTER THE VALUE OF ELEMENTS : 566
ENTER THE VALUE OF ELEMENTS : 786
ENTER THE VALUE OF ELEMENTS : 555
ENTER THE VALUE OF ELEMENTS : 4567
ENTER THE VALUE OF ELEMENTS : 26
ENTER THE VALUE OF ELEMENTS : 18
ENTER THE VALUE OF ELEMENTS : 4503
LARGEST ELEMENT IS :  4567
```

# PRACTICAL 2

**2[a]**

**AIM : Write a program to create a dictionary where keys represent products and values represent prices. Find out and display products with the minimum and maximum prices.**

**INPUT :**

```
dict = {'AB': 100, 'CD': 500, 'EF' :1000}
print("Maximum")
a = max(dict.values())
b = max(dict ,key=dict.get)
print(b,"=",a)
print("       ")
print("Minimum")
c = min(dict.values())
d = min(dict ,key=dict.get)
print(d,"=",c)
```

**OUTPUT :**

```
Maximum
EF = 1000

Minimum
AB = 100
```

**2[b]**

**AIM : Write a program to create two dictionaries with numerical values of each key and merge the dictionaries such that values of common keys should be added.**

**INPUT :**

```
def Merge(dict1, dict2):
    return(dict2.update(dict1))
dict1 = {'a': 10, 'b': 8}
dict2 = {'b': 6, 'c': 4}
for i in dict2:
    if i in dict1:
        dict2[i] = dict1[i] + dict2[i]
```

```
    else:
       pass

for i in dict1:
    if i not in dict2:
        dict2[i] = dict1[i]

print(dict2)
```

**OUTPUT :**

```
{'b': 14, 'c': 4, 'a': 10}
```

**2[c]**

**AIM : Write a program to create a dictionary where keys represent characters in a string and values represent their frequencies. For example,**

**Input: "adani" ; Expected output: {'a': 2, 'd': 1, 'i': 1, 'n': 1}**

**INPUT :**

```
str = input("Enter the string: ")
A = list()
for i in str:
   print(i)
   A.append(i)
print(A)
sub = {}
for i in A:
  sub[i] = sub.get(i,0) + 1
print(sub)
```

**OUTPUT :**

```
Enter the string: adani
a
d
a
n
i
['a', 'd', 'a', 'n', 'i']
{'a': 2, 'd': 1, 'n': 1, 'i': 1}
```

**2[d]**

**AIM : Write a program to create two sets and find out union, intersection, and symmetric difference between them.**

**INPUT :**

```
set1 = {"A","B","C","N"}
set2 = {"M","N","O","P"}

#UNION
uni = set1.union(set2)
print("UNION : ", uni)
print(" ")

#INTERSECTION
intr = set1.intersection(set2)
print("INTERSECTION : ",intr)
print(" ")

#SYMMETRIC DIFFERNCE
sym = (set1).symmetric_difference(set2)
print("SYMMETRIC DIFFERNCE : ",sym)
print(" ")
```

**OUTPUT :**

```
UNION :  {'C', 'O', 'A', 'M', 'P', 'B', 'N'}

INTERSECTION :  {'N'}

SYMMETRIC DIFFERNCE :  {'C', 'O', 'A', 'M', 'P', 'B'}
```

**2[e]**

**AIM : Write a program to check whether the number entered by the user is a perfect number.**

**INPUT:**

```
num=int(input("Enter the number: "))
sum=0

for i in range(1,num):
    if (num%i==0):
        sum=sum+i

if(sum==num):
    print("The entered number is a perfect number")
else:
    print("The entered number is not a perfect number")
```

**OUTPUT :**

( i ) Prefect Number

```
Enter the number: 6
The entered number is a perfect number
```

( ii ) Not A Prefect Number

```
Enter the number: 9
The entered number is not a perfect number
```

# PRACTICAL 3

**3[a]**

**AIM : Write a program to take inputs from the user and return a list with unique elements.**

**INPUT :**

```
a = True
lst = list();

while(a):
    n = input("Enter Input : ")
    if(n in lst):
        print("This element is already in the list !")
        continue
    else:
        lst.append(n)
    b = input("You want to continue (Y/N) ").capitalize()

    if(b == 'Y'):
        a = True
    elif(b == 'N'):
         a = False

print("List :- ",lst)
```

**OUTPUT :**

```
Enter Input : NEEL
You want to continue (Y/N) Y
Enter Input : PATEL
You want to continue (Y/N) Y
Enter Input : NEEL
This element is already in the list !
Enter Input : PRACT3
You want to continue (Y/N) N
List :-  ['NEEL', 'PATEL', 'PRACT3']
```

**3[b]**

**AIM : Write a program to check the validity of a password entered by the user. The password should contain the following specifications for validation.**

- **At least one lowercase and one uppercase alphabets.**
- **At least one digit.**
- **At least one special character @, #, or $.**
- **Minimum length of 6 characters.**
- **Maximum length of 12 characters.**

**INPUT :**

```
password=input("Enter the password: ")
p=False
ls=list()
Upper=0
Lower=0
digit =0
special_character=0

for i in password:
    ls.append(i)
    if(i.isdigit() is True):
       digit+=1
    elif(i.isupper() is True):
       Upper+=1
    elif(i.islower() is True):
       Lower+=1
    else:
       special_character+=1
while(p!=True):
 if(len(password)>=6 and len(password)<=12 and digit>0 and Upper>0 and
Lower>0
and special_character>0 ):
   print("Password is correct")
   p=True
 else:
   password = input("Enter the password the again: ")
```

**OUTPUT :**

- **Valid Password :**

```
Enter the password: Abc@12345
Password is correct
```

- **Invalid Password :**

```
Enter the password: pract@345
Enter the password the again:
```

**3[c]**

**AIM :Write a program to print the following pattern using a nested loop.**

**1**

**22**

**333**

**4444**

**55555**

**666666**

**7777777**

**88888888**

**999999999**

**INPUT :**

```
for i in range(1,10):
    for j in range(1,i+1):
       print(i,end="")
    print()
```

**OUTPUT :**

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
```

**3[d]**

**AIM : Write a program with "Circle" class with an argument asking for a radius value. Build two methods to compute area and perimeter of a circle using radius. Also, create two instances "c1" and "c2" of class "Circle" for the following tasks.**

- **Call both the methods using both instances to print corresponding outputs.**
- **Modifyadius value in "c2" instance. Compute area or perimeter by calling appropriate method.**

**INPUT :**

```
pi=3.14
class Circle():
   def __init__(self,r):
     self.r = r
   def area(self):
      return pi*self.r*self.r
   def perimeter(self,rad):
      return 2*pi*rad
c1 = Circle(12)
c2 = Circle(17)
c = c1.area()
print("The area of c1 is ",c)
c = c1.perimeter(6)
print("\n The perimeter of c1 is ",c)
c = c2.area()
print("\n The area of c2 is ", c)
c = c2.perimeter(9)
print("\n The perimeter of c2 is ",c)
```

**OUTPUT :**

```
The area of c1 is  452.15999999999997

 The perimeter of c1 is  37.68

 The area of c2 is  907.46

 The perimeter of c2 is  56.52
```

**3[e]**

**AIM : Write a program with "ComplexNumber" class which containsreal and imaginary number attributes. Create "n1" and "n2" instances for the following tasks.**

- **Create a method to add "n1" and "n2".**
- **Create a method to calculate modulus of "n1" and "n2".**
- **Delete the imaginary part of instance "n2".**
- **Check whether "n1" and "n2" have imaginary parts.**

**INPUT:**

```
import math
class ComplexNumber():
  def __init__(self,real,imaginary):
   self.real = real
   self.imaginary = imaginary
  def add(self,n1,n2):
    print("The real part is: " ,n1.real+n2.real)
    print("The imaginary part is: ",n1.imaginary+n2.imaginary)
  def modulus(self):
    print(math.sqrt((self.real*self.real)+(self.imaginary*self.imaginar
y)))
  def delete(self):
    self.imaginary=None
    print("The imaginary part is deleted")
  def Check_Imaginary(self):
    if self.imaginary!=None:
       print("The imaginary part is present")
    else:
       print("The imaginary part is not present")
n1=ComplexNumber(5,8)
n2=ComplexNumber(91,73)
print("n1,n2 after addition: ")
n1.add(n1,n2)
print("The modulus for n1 is: ")
n1.modulus()
print("Deleting imaginary part for n2: ")
n2.delete()
print("Checking n1's imaginary part: ")
n1.Check_Imaginary()
print("Checking n2's imaginary part: ")
n2.Check_Imaginary()
```

**OUTPUT :**

```
n1,n2 after addition:
The real part is:  96
The imaginary part is:  81
The modulus for n1 is:
9.433981132056603
Deleting imaginary part for n2:
The imaginary part is deleted
Checking n1's imaginary part:
The imaginary part is present
Checking n2's imaginary part:
The imaginary part is not present
```

# PRACTICAL 4

**4[a]**

**AIM : Write a program to calculate element-wise difference between the neighbouring elements of a given array.**

**INPUT :**

```
import numpy as np
a = np.array([1,14,16,19,24,56,78,99])
print("Original array: ",a)
print("Difference between adjacent elements : ",np.diff(a))
```

**OUTPUT :**

```
Original array:  [ 1 14 16 19 24 56 78 99]
Difference between adjacent elements :  [13  2  3  5 32 22 21]
```

**4[b]**

**AIM : Write a program using NumPy to multiply two user-defined matrices of size n*n where n > 2.**

**INPUT :**

```
import numpy as np
n = int(input("Enter the value of n for n x n matrix : "))
b = np.random.randint(1,9,size=(n,n))
print("\nMATRIX 1 : " , "\n",b)
c = np.random.randint(1,9,size=(n,n))
print("\nMATRIX 2 : ","\n" ,c)
d = np.dot(b,c)
print("\nMULTIPLICATION OF MATRIX 1 AND MATRIX 2 IS : ", "\n",d)
```

**OUTPUT :**

```
Enter the value of n for n x n matrix : 5

MATRIX 1 :
 [[1 3 4 2 8]
 [6 7 2 1 7]
 [8 1 3 8 4]
 [3 8 4 2 8]
 [6 5 7 1 2]]

MATRIX 2 :
 [[4 6 8 6 7]
 [1 2 6 4 2]
 [2 8 6 6 5]
 [5 5 5 4 6]
 [2 5 6 2 2]]

MULTIPLICATION OF MATRIX 1 AND MATRIX 2 IS :
 [[ 41  94 108  66  61]
 [ 54 106 149  94  86]
 [ 87 134 152 110 129]
 [ 54 116 154  98  85]
 [ 52 117 137 106  97]]
```

**4[c]**

**AIM : Write a program to take a 3*3 array (matrix) from the userand calculate its transpose as well as determinant. Also, find whether the matrix is invertible; if yes, find the inverse of the matrix.**

**INPUT :**

```
n = int(input("Enter the size of matrix : "))
e = [[int(input()) for j in range(n)]for i in range (n)]
print("\nTHE MATRIX FOR ",n ,"X",n , "is : ", "\n",e)
f = np.array(e)
print("\nTRANSPOSE OF MATRIX IS : ")
print(f.transpose())
det = np.linalg.det(f)
print("\nDETERMINANT OF MATRIX IS : ","\n", det)
if (det==0):
  print("\nMATRIX IS NOT INVERTIBLE")
else:
  print("\nMATRIX IS INVERTIBLE & INVERSE IS:")
  print(np.linalg.inv(e))
```

**OUTPUT :**

```
Enter the size of matrix : 2
45
23
97
16

THE MATRIX FOR  2 X 2 is :
 [[45, 23], [97, 16]]

TRANSPOSE OF MATRIX IS :
[[45 97]
 [23 16]]

DETERMINANT OF MATRIX IS :
 -1510.999999999999

MATRIX IS INVERTIBLE & INVERSE IS:

 [[-0.01058901  0.01522171]
 [ 0.0641959  -0.0297816 ]]
```

**4[d]**

**AIM : Write a program to solve three linear equations with variables x, y, and z using NumPy.**

**INPUT :**

```
import numpy as np
g = np.array([[2,3,4],[5,-4,9],[3,6,8]])
print("THE COEFFICIENT MATRIX :","\n",g)
h = np.array([24,57,43])
print("\nMATTRIX OF CONSTANTS : ","\n",h)
l = np.linalg.inv(g).dot(h)
print("\nSOLUTION : ""\n[a,b,c] = ",l)
```

**OUTPUT :**

```
THE COEFFICIENT MATRIX :
 [[ 2  3  4]
 [ 5 -4  9]
 [ 3  6  8]]

MATTRIX OF CONSTANTS :
 [24 57 43]

SOLUTION :
[a,b,c] =  [ 5.         -0.04651163  3.53488372]
```

**4[e]**

**AIM : Write a program to create a 5*5 array with random values and find the minimum and maximum values using NumPy.**

**INPUT:**

```
import numpy as np
n = int(input("ENTER SIZE OF MATRIX : "))
m = np.random.randint(1,20,size=(n,n))
print("\nMATRIX IS : ","\n" , m)
print("\nTHE MAXIMUM VALUE IS = ",m.max())
print("\nTHE MINIMUM VALUE IS = ",m.min())
```

**OUTPUT :**

```
ENTER SIZE OF MATRIX : 5

MATRIX IS :
 [[19 19  4 18  2]
 [ 5 10 14  4  3]
 [19  9 10  2 18]
 [14 10 19  3 10]
 [ 2  5 19 15 13]]

THE MAXIMUM VALUE IS =  19

THE MINIMUM VALUE IS =  2
```

# PRACTICAL 5

**5[a]**

**AIM : Write a program to create a dataframe "student_data" using pandas with the following details; provide at least 10 sample data and display the same. Student name, enrollment number, date of birth, city, CPI**

**INPUT :**

```
import pandas as pd
dt = {"NAME":["NEEL","PREET","KAVYA","VARSHIT","DEVAM","HARI","HARSH","
MANTHAN","DHAMO","ARYA"],
      "ENROLLMENT NO.":[28,24,3,9,14,19,16,25,13,6],
      "D.O.B.":['28 JAN 2003','24 AUG 2002','03 SEPT 2002','09 MAR 2003
','14 JAN 2001','09 JUL 2003','16 NOV 2003','24 SEPT 2003','13 JUL 2003
','21 OCT 2002'],
      "CITY":['Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad','Ahemdaba
d','Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad'],
      "CPI":[9,8,9,8,9,8,8,9,8,9]}
labels=[1,2,3,4,5,6,7,8,9,10]
Student_data = pd.DataFrame(dt,index=labels)
print(Student_data)
```

**OUTPUT :**

| | NAME | ENROLLMENT NO. | D.O.B. | CITY | CPI |
|---|---|---|---|---|---|
| 1 | NEEL | 28 | 28 JAN 2003 | Ahemdabad | 9 |
| 2 | PREET | 24 | 24 AUG 2002 | Ahemdabad | 8 |
| 3 | KAVYA | 3 | 03 SEPT 2002 | Ahemdabad | 9 |
| 4 | VARSHIT | 9 | 09 MAR 2003 | Ahemdabad | 8 |
| 5 | DEVAM | 14 | 14 JAN 2001 | Ahemdabad | 9 |
| 6 | HARI | 19 | 09 JUL 2003 | Ahemdabad | 8 |
| 7 | HARSH | 16 | 16 NOV 2003 | Ahemdabad | 8 |
| 8 | MANTHAN | 25 | 24 SEPT 2003 | Ahemdabad | 9 |
| 9 | DHAMO | 13 | 13 JUL 2003 | Ahemdabad | 8 |
| 10 | ARYA | 6 | 21 OCT 2002 | Ahemdabad | 9 |

**5[b]**

**AIM : Write a program to select and display student names and CPI columns from "student_data".**

**INPUT :**

```
Student_data.loc[:,['NAME','CPI']]
```

**OUTPUT :**

| | NAME | CPI |
|---|---|---|
| 1 | NEEL | 9 |
| 2 | PREET | 8 |
| 3 | KAVYA | 9 |
| 4 | VARSHIT | 8 |
| 5 | DEVAM | 9 |
| 6 | HARI | 8 |
| 7 | HARSH | 8 |
| 8 | MANTHAN | 9 |
| 9 | DHAMO | 8 |
| 10 | ARYA | 9 |

**5[c]**

**AIM : Write a program to sort "student_data" using CPI column in descending order; further, sort the same using student name column in ascending order and display it.**

**INPUT :**

```
Student = Student_data.sort_values(by=['CPI','NAME'],ascending=[False,
True])
Student
```

**OUTPUT :**

| | NAME | ENROLLMENT NO. | D.O.B. | CITY | CPI | Blood_group |
|---|---|---|---|---|---|---|
| 10 | ARYA | 6 | 21 OCT 2002 | Ahemdabad | 9 | B+ |
| 5 | DEVAM | 14 | 14 JAN 2001 | Ahemdabad | 9 | O+ |
| 3 | KAVYA | 3 | 03 SEPT 2002 | Ahemdabad | 9 | A+ |
| 8 | MANTHAN | 25 | 24 SEPT 2003 | Ahemdabad | 9 | B- |
| 1 | NEEL | 28 | 28 JAN 2003 | Ahemdabad | 9 | AB+ |
| 9 | DHAMO | 13 | 13 JUL 2003 | Ahemdabad | 8 | O- |
| 6 | HARI | 19 | 09 JUL 2003 | Ahemdabad | 8 | O- |
| 7 | HARSH | 16 | 16 NOV 2003 | Ahemdabad | 8 | B+ |
| 2 | PREET | 24 | 24 AUG 2002 | Ahemdabad | 8 | AB- |
| 4 | VARSHIT | 9 | 09 MAR 2003 | Ahemdabad | 8 | A- |

**5[d]**

**AIM : Write a program to delete city column from "student_data" and display the remaining dataframe.**

**INPUT :**

```
Student_data.drop('CITY',axis=1)
```

**OUTPUT :**

| | NAME | ENROLLMENT NO. | D.O.B. | CPI |
|---|---|---|---|---|
| 1 | NEEL | 28 | 28 JAN 2003 | 9 |
| 2 | PREET | 24 | 24 AUG 2002 | 8 |
| 3 | KAVYA | 3 | 03 SEPT 2002 | 9 |
| 4 | VARSHIT | 9 | 09 MAR 2003 | 8 |
| 5 | DEVAM | 14 | 14 JAN 2001 | 9 |
| 6 | HARI | 19 | 09 JUL 2003 | 8 |
| 7 | HARSH | 16 | 16 NOV 2003 | 8 |
| 8 | MANTHAN | 25 | 24 SEPT 2003 | 9 |
| 9 | DHAMO | 13 | 13 JUL 2003 | 8 |
| 10 | ARYA | 6 | 21 OCT 2002 | 9 |

**5[e]**

**AIM : Write a program to insert a new column.**

**INPUT:**

```
Blood_group = ['AB+','AB-','A+','A-','O+','O-','B+','B-','O-','B+']
Student_data['Blood_group'] = Blood_group
Student_data
```

**OUTPUT :**

| | NAME | ENROLLMENT NO. | D.O.B. | CITY | CPI | Blood_group |
|---|---|---|---|---|---|---|
| 1 | NEEL | 28 | 28 JAN 2003 | Ahemdabad | 9 | AB+ |
| 2 | PREET | 24 | 24 AUG 2002 | Ahemdabad | 8 | AB- |
| 3 | KAVYA | 3 | 03 SEPT 2002 | Ahemdabad | 9 | A+ |
| 4 | VARSHIT | 9 | 09 MAR 2003 | Ahemdabad | 8 | A- |
| 5 | DEVAM | 14 | 14 JAN 2001 | Ahemdabad | 9 | O+ |
| 6 | HARI | 19 | 09 JUL 2003 | Ahemdabad | 8 | O- |
| 7 | HARSH | 16 | 16 NOV 2003 | Ahemdabad | 8 | B+ |
| 8 | MANTHAN | 25 | 24 SEPT 2003 | Ahemdabad | 9 | B- |
| 9 | DHAMO | 13 | 13 JUL 2003 | Ahemdabad | 8 | O- |
| 10 | ARYA | 6 | 21 OCT 2002 | Ahemdabad | 9 | B+ |

# PRACTICAL 6

**6[a]**

**AIM : Create a .txt file with at least ten lines and write a program to perform the following tasks.**

- **Stream the data in the file.**
- **Display text at the odd numbered lines in the file.**
- **Randomly sample 30% lines and display.**
- **Read the file using pandas for parsing.**

**INPUT :**

- **Stream the data in the file.**

```
with open('Python.txt') as f:
  data = f.read()
  print(data)
with open('Python.txt','a') as f:
  f.write('Hello Programmer')
```

- **Display text at the odd numbered lines in the file.**

```
fd = open('Python.txt','r')
cont = fd.readlines()
for i in range(len(cont)):
   if(i%2 == 0):
      print(cont[i])
```

- **Randomly sample 30% lines and display.**

```
import random
fd = open('Python.txt','r')
cont = fd.readlines()
lst = list()
```

```
for i in range(len(cont)):
   lst.append(cont[i])
ln = len(lst)
lnt = 0.3*ln
print("Three random lines are: ")
print(random.sample(lst,int(lnt))
```

- **Read the file using pandas for parsing.**

```
import pandas as pd
df = pd.read_csv("Python.txt",sep=".")
print(df)
```

**OUTPUT :**

```
Python is an open source programming language.
It was made to be easy-to-read-and-understand and powerful.
A Dutch programmer named Guido van Rossum made Python in 1991.
He named it after the television program Monty Python's Flying Circus.
Many Python examples and tutorials include jokes from the show.
Python is an interpreted language.
Interpreted languages do not need to be compiled to run.
A program called an interpreter runs Python code on almost any kind of computer.
This means that a programmer can change the code and quickly see the results.
This also means Python is slower than a compiled language like C, because it is not machine code directly.
Python has become one of the most famous programming languages on the world as of late.
It's utilized in all that from AI to building sites and programming testing.
It tends to be utilized by engineers and non-designers the same.
Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.
```

```
Python is an open source programming language.

A Dutch programmer named Guido van Rossum made Python in 1991.

Many Python examples and tutorials include jokes from the show.

Interpreted languages do not need to be compiled to run.

This means that a programmer can change the code and quickly see the results.

Python has become one of the most famous programming languages on the world as of late.

It tends to be utilized by engineers and non-designers the same.
```

```
Three random lines are:
["It's utilized in all that from AI to building sites and programming testing. \n", 'Interpreted languages
do not need to be compiled to run. \n', 'This means that a programmer can change the code and quickly see
the results. \n', 'It tends to be utilized by engineers and non-designers the same.\n']
```

```
        Python is an open source programming language        Unnamed: 1
0   It was made to be easy-to-read-and-understand ...               NaN
1   A Dutch programmer named Guido van Rossum made...
2   He named it after the television program Monty...
3   Many Python examples and tutorials include jok...               NaN
4                  Python is an interpreted language
5   Interpreted languages do not need to be compil...
6   A program called an interpreter runs Python co...
7   This means that a programmer can change the co...
8   This also means Python is slower than a compil...               NaN
9   Python has become one of the most famous progr...
10  It's utilized in all that from AI to building ...
11  It tends to be utilized by engineers and non-d...               NaN
12  Python drew inspiration from other programming...  Hello Programmer
```

**6[b]**

**AIM : Create or download a .csv file from the publically available datasets and write a program to perform the following tasks.**

- **Read the file using pandas for parsing.**
- **Display any two columns from the file.**
- **Display any two records (rows) from the file.**

**INPUT :**

- **Read the file using pandas for parsing.**

```
import pandas as pd
df = pd.read_csv('student.csv')
df
```

- **Display any two columns from the file.**

```
df_new = df[['name','class']]
 print(df_new)
```

- **Display any two records (rows) from the file.**

```
 print(df.iloc[2:4])
```

**OUTPUT :**

| | id | name | class | mark | gender |
|---|---|---|---|---|---|
| 0 | 1 | John Deo | Four | 75 | female |
| 1 | 2 | Max Ruin | Three | 85 | male |
| 2 | 3 | Arnold | Three | 55 | male |
| 3 | 4 | Krish Star | Four | 60 | female |
| 4 | 5 | John Mike | Four | 60 | female |
| 5 | 6 | Alex John | Four | 55 | male |
| 6 | 7 | My John Rob | Fifth | 78 | male |
| 7 | 8 | Asruid | Five | 85 | male |
| 8 | 9 | Tes Qry | Six | 78 | male |
| 9 | 10 | Big John | Four | 55 | female |
| 10 | 11 | Ronald | Six | 89 | female |
| 11 | 12 | Recky | Six | 94 | female |
| 12 | 13 | Kty | Seven | 88 | female |
| 13 | 14 | Bigy | Seven | 88 | female |

| | name | class |
|---|---|---|
| 0 | John Deo | Four |
| 1 | Max Ruin | Three |
| 2 | Arnold | Three |
| 3 | Krish Star | Four |
| 4 | John Mike | Four |
| 5 | Alex John | Four |
| 6 | My John Rob | Fifth |
| 7 | Asruid | Five |
| 8 | Tes Qry | Six |
| 9 | Big John | Four |
| 10 | Ronald | Six |
| 11 | Recky | Six |
| 12 | Kty | Seven |
| 13 | Bigy | Seven |
| 14 | Tade Row | Four |
| 15 | Gimmy | Four |
| 16 | Tumyu | Six |
| 17 | Honny | Five |
| 18 | Tinny | Nine |
| 19 | Jackly | Nine |
| 20 | Babby John | Four |
| 21 | Reggid | Seven |
| 22 | Herod | Eight |
| 23 | Tiddy Now | Seven |

```
   id        name  class  mark  gender
2   3      Arnold  Three    55    male
3   4  Krish Star   Four    60  female
```

**6[c]**

**AIM : Create or download a .xls file from the publically available datasets with at least three worksheets and write a program to perform the following tasks using pandas for parsing.**

- **Read sheet2 from .xls and display it.**
- **Read specific columns from the file and display them.**
- **Add a column in sheet2 of the file and fill it with NaN values.**

**INPUT :**

- **Read sheet2 from .xls and display it.**

```
df3 = pd.read_excel('student.xlsx',sheet_name='sheet2')
df3
```

- **Read specific columns from the file and display them.**

```
df3['name']
```

- **Add a column in sheet2 of the file and fill it with NaN values.**

```
import numpy as np
df3['new'] = np.nan
df3
```

**OUTPUT :**

| | id | name | class | mark | gender |
|---|---|---|---|---|---|
| 0 | 1 | John Deo | Four | 75 | female |
| 1 | 2 | Max Ruin | Three | 85 | male |
| 2 | 3 | Arnold | Three | 55 | male |
| 3 | 4 | Krish Star | Four | 60 | female |
| 4 | 5 | John Mike | Four | 60 | female |
| 5 | 6 | Alex John | Four | 55 | male |
| 6 | 7 | My John Rob | Five | 78 | male |
| 7 | 8 | Asruid | Five | 85 | male |
| 8 | 9 | Tes Qry | Six | 78 | male |
| 9 | 10 | Big John | Four | 55 | female |
| 10 | 11 | Ronald | Six | 89 | female |
| 11 | 12 | Recky | Six | 94 | female |
| 12 | 13 | Kty | Seven | 88 | female |
| 13 | 14 | Bigy | Seven | 88 | female |
| 14 | 15 | Tade Row | Four | 88 | male |

```
0         John Deo
1        Max Ruin
2          Arnold
3      Krish Star
4       John Mike
5       Alex John
6     My John Rob
7          Asruid
8         Tes Qry
9        Big John
10         Ronald
11          Recky
12            Kty
13           Bigy
14       Tade Row
15          Gimmy
16          Tumyu
17          Honny
18          Tinny
19         Jackly
20     Babby John
21         Reggid
```

| | id | name | class | mark | gender | new |
|---|---|---|---|---|---|---|
| 0 | 1 | John Deo | Four | 75 | female | NaN |
| 1 | 2 | Max Ruin | Three | 85 | male | NaN |
| 2 | 3 | Arnold | Three | 55 | male | NaN |
| 3 | 4 | Krish Star | Four | 60 | female | NaN |
| 4 | 5 | John Mike | Four | 60 | female | NaN |
| 5 | 6 | Alex John | Four | 55 | male | NaN |
| 6 | 7 | My John Rob | Five | 78 | male | NaN |
| 7 | 8 | Asruid | Five | 85 | male | NaN |
| 8 | 9 | Tes Qry | Six | 78 | male | NaN |
| 9 | 10 | Big John | Four | 55 | female | NaN |
| 10 | 11 | Ronald | Six | 89 | female | NaN |
| 11 | 12 | Recky | Six | 94 | female | NaN |
| 12 | 13 | Kty | Seven | 88 | female | NaN |
| 13 | 14 | Bigy | Seven | 88 | female | NaN |
| 14 | 15 | Tade Row | Four | 88 | male | NaN |
| 15 | 16 | Gimmy | Four | 88 | male | NaN |
| 16 | 17 | Tumyu | Six | 54 | male | NaN |
| 17 | 18 | Honny | Five | 75 | male | NaN |

**6[d]**

**AIM : Write a program to perform the following tasks using an image from online source using skimage.**

- **Display the original image.**
- **Display the image in gray scale.**
- **Find the shape and type of the image.**
- **Display a cropped version of the image.**
- **Display a resized version of the image.**

**INPUT:**

- **Display the original image.**

```
import os
from skimage.io import imshow,imread
eimg = "https://i.pinimg.com/736x/bb/08/c8/bb08c890c51d1af38185a40ddea7
cf92.jpg"
img = imread(eimg)
print(img.shape)
imshow(img)
imshow(img)
```

- **Display the image in gray scale.**

```
import os
from skimage.io import imshow,imread
eimg = "https://i.pinimg.com/736x/bb/08/c8/bb08c890c51d1af38185a40ddea7
cf92.jpg"
img = imread(eimg,as_gray = True)
print(img.shape)
imshow(img)
```

- **Find the shape and type of the image.**

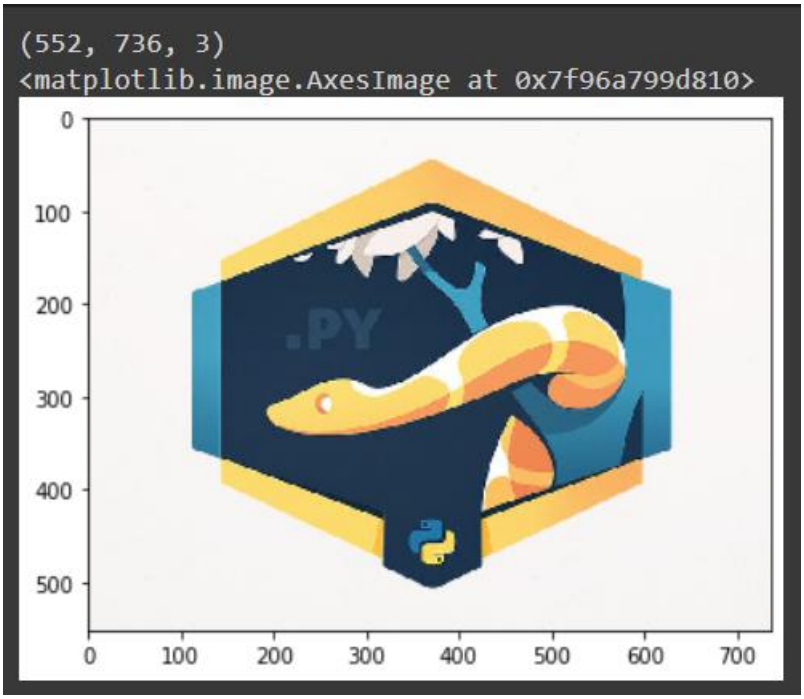
```
print(img.shape)
print(type(img))
```

- **Display a cropped version of the image.**

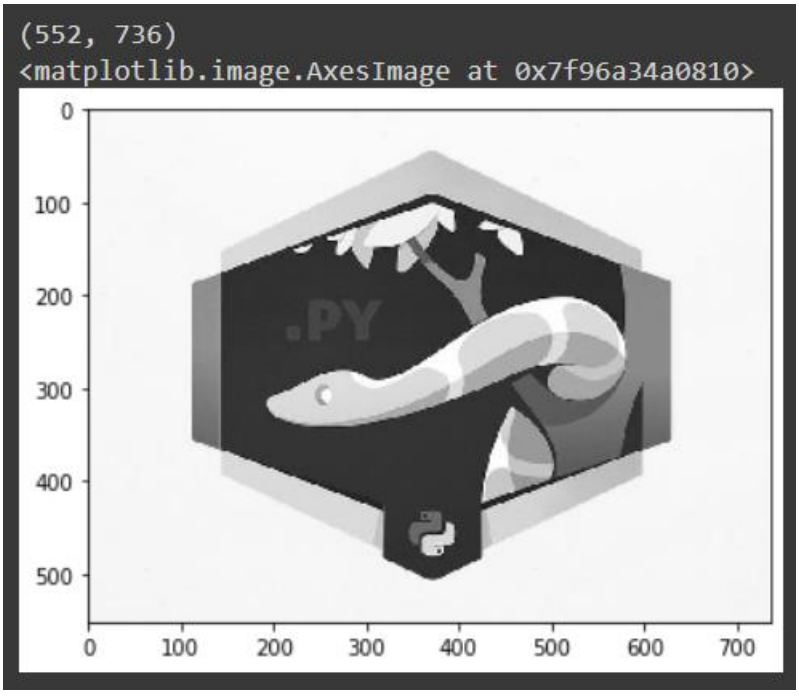
```
import matplotlib.pyplot as plt
cropped = img[100:(img.shape[0]-100),100:(img.shape[1]-100)]
# imshow(img)
imshow(cropped)
```

**OUTPUT :**

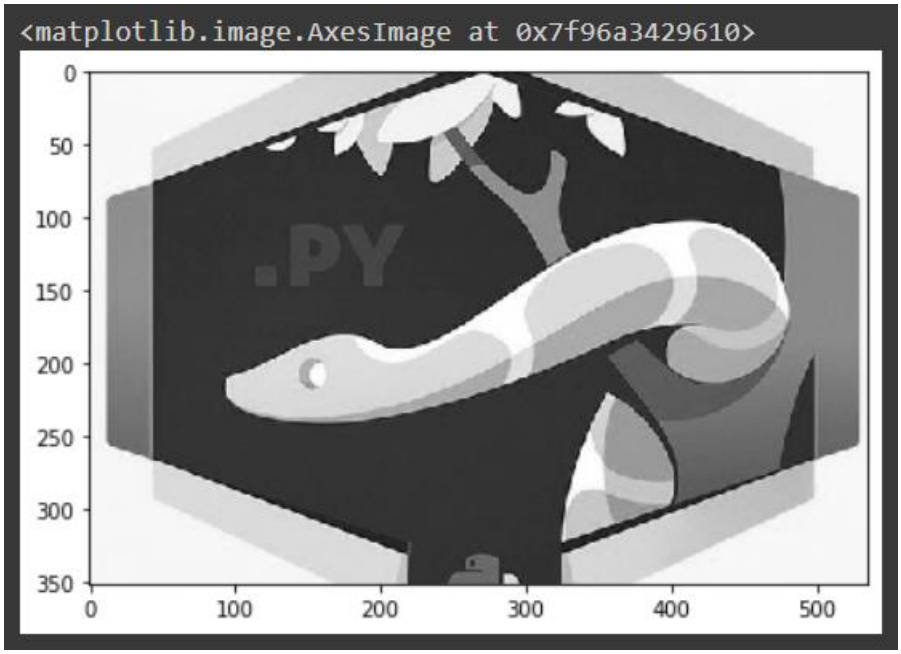
```
(552, 736, 3)
<matplotlib.image.AxesImage at 0x7f96a799d810>
```




```
(552, 736)
<matplotlib.image.AxesImage at 0x7f96a34a0810>
```




```
(338, 600)
<class 'numpy.ndarray'>
```


<matplotlib.image.AxesImage at 0x7f96a3429610>
0
50
100
150
200
250
300
350
0
100
200
300
400
500

## PRACTICAL 7

**7[a]**

**AIM : List down the features of matplotlib.**

**INPUT :**

**1. Easy to Code**

**Python is a very high-level programming language, yet it is effortless to learn. Anyone can learn to code in Python in just a few hours or a few days. Mastering Python and all its advanced concepts, packages and modules might take some more time. However, learning the basic Python syntax is very easy, as compared to other popular languages like C, C++, and Java.**

**2. Easy to Read**

**Python code looks like simple English words. There is no use of semicolons or brackets, and the indentations define the code block. You can tell what the code is supposed to do simply by looking at it.**

**3. Free and Open-Source**

**Python is developed under an OSI-approved open source license. Hence, it is completely free to use, even for commercial purposes. It doesn't cost anything to download Python or to include it in your application. It can also be freely modified and re-distributed. Python can be downloaded from the official Python website.**

**4. Robust Standard Library**

**Python has an extensive standard library available for anyone to use. This means that programmers don't have to write their code for every single thing unlike other programming languages. There are libraries for image manipulation, databases, unit-testing, expressions and a lot of other functionalities. In addition to the standard**

**library, there is also a growing collection of thousands of components, which are all available in the Python Package Index.**

**5. Interpreted**

**When a programming language is interpreted, it means that the source code is executed line by line, and not all at once. Programming languages such as C++ or Java are not interpreted, and hence need to be compiled first to run them. There is no need to compile Python because it is processed at runtime by the interpreter.**

**6. Portable**

**Python is portable in the sense that the same code can be used on different machines. Suppose you write a Python code on a Mac. If you want to run it on Windows or Linux later, you don't have to make any changes to it. As such, there is no need to write a program multiple times for several platforms.**

**7. Object-Oriented and Procedure-Oriented**

**A programming language is object-oriented if it focuses design around data and objects, rather than functions and logic. On the contrary, a programming language is procedure-oriented if it focuses more on functions (code that can be reused). One of the critical Python features is that it supports both object-oriented and procedure-oriented programming.**

**8. Extensible**

**A programming language is said to be extensible if it can be extended to other languages. Python code can also be written in other languages like C++, making it a highly extensible language.**

**9. Expressive**

**Python needs to use only a few lines of code to perform complex tasks. For example, to display Hello World, you simply need to type one line - print(“Hello World”). Other languages like Java or C would take up multiple lines to execute this.**

**10. Support for GUI**

**One of the key aspects of any programming language is support for GUI or Graphical User Interface. A user can easily interact with the software using a GUI. Python offers various toolkits, such as Tkinter, wxPython and JPython, which allows for GUI's easy and fast development.**

**11. Dynamically Typed**

**Many programming languages need to declare the type of the variable before runtime. With Python, the type of the variable can be decided during runtime. This makes Python a dynamically typed language.**

**For example, if you have to assign an integer value 20 to a variable “x”, you don’t need to write int x = 20. You just have to write x = 15.**

**12. High-level Language**

**Python is a high-level programming language because programmers don’t need to remember the system architecture, nor do they have to manage the memory. This makes it super programmer-friendly and is one of the key features of Python.**

**13. Simplify Complex Software Development**

**Python can be used to develop both desktop and web apps and complex scientific and numerical applications. Python's data analysis features help you create custom big data solutions without so much time and effort. You can also use the Python data visualization libraries and APIs to present data in a more appealing way. Several**

**advanced software developers use Python to accomplish high-end AI and natural language processing tasks.**

**14. Other Advanced Programming Features**

**Python contains several advanced programming features such as generators (used to create iterators with a different approach that most other languages) and list comprehensions (used to create new lists from other iterables). Python also has automatic memory management eliminating the need to manually allocate and free memory in the code.**

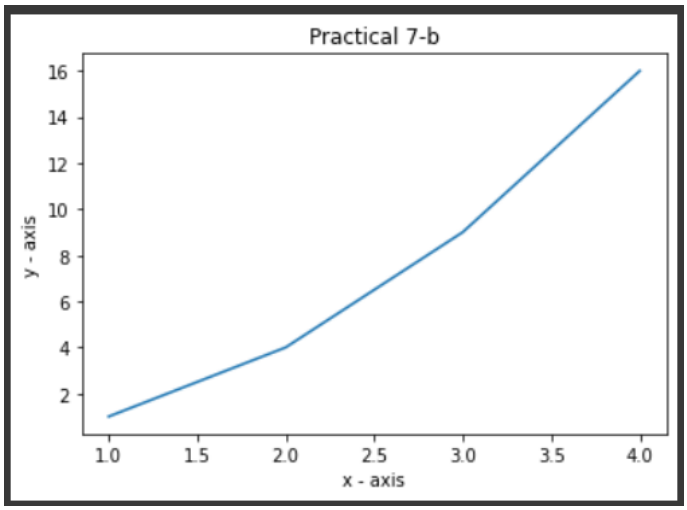
**7[b]**

**AIM : Write a program to display a line graph with x = [1, 2, 3, 4] and y = [1, 4, 9, 16]. Give appropriate labels to both the axes.**

**INPUT :**

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [1,4,9,16]
plt.plot(x, y)
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.title('Practical 7-b')
plt.show()
```

**OUTPUT :**



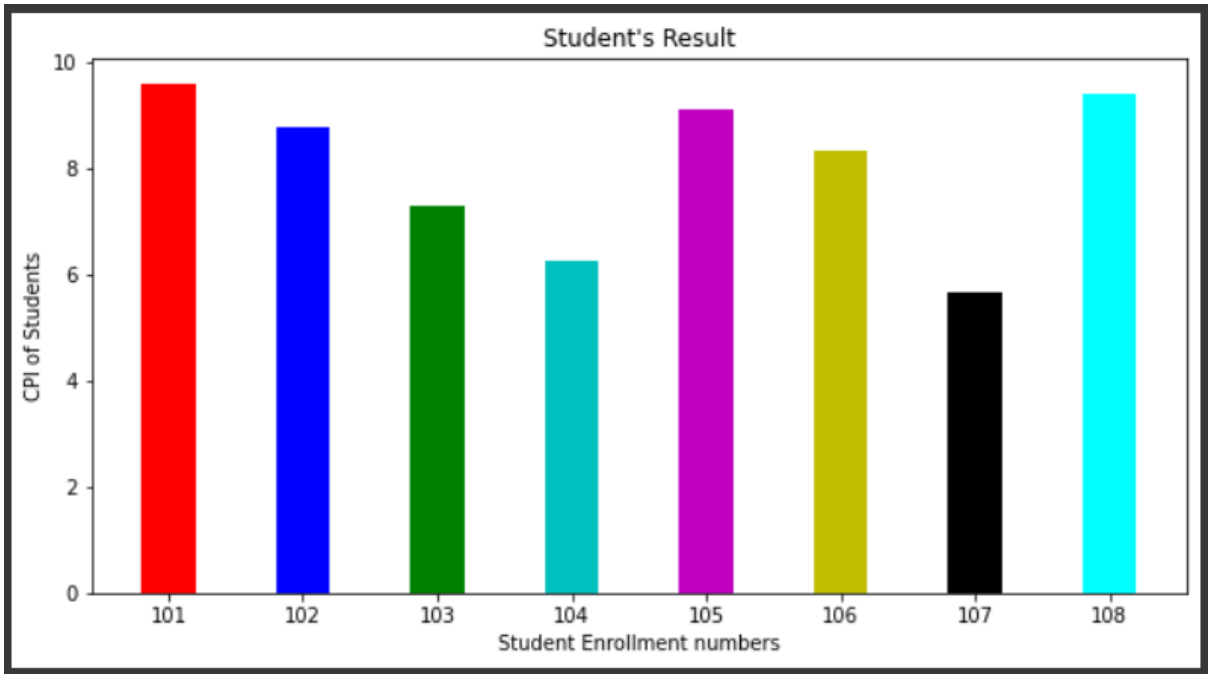
**7[c]**

**AIM : Write a program to display a bar chart where x-axis represents student enrolment numbers and y-axis represents CPI. Give appropriate labels to both the axes.**

**INPUT :**

```
import numpy as np
import matplotlib.pyplot as plt
data = {'101':9.58, '102':8.78, '103':7.30,'104':6.25,'105':9.11, '106'
:8.32, '107':5.67, '108':9.40}
courses = list(data.keys())
values = list(data.values())
fig = plt.figure(figsize = (10, 5))
c = [ 'r','b','g','c','m','y','k','cyan']
plt.bar(courses, values, color =c, width = 0.4)
plt.xlabel("Student Enrollment numbers")
plt.ylabel("CPI of Students")
plt.title("Student's Result")
plt.show()
```

**OUTPUT :**



## PRACTICAL 8

**8[a]**

**AIM : Write a program to display a pie chart for movie type distribution. The data includes Comedy (4), Action (5), Drama (7), and Sci-Fi (4).**

**INPUT :**

```
import matplotlib.pyplot as plt
x = [4,5,7,4]
l = ['COMEDY','ACTION','DRAMA','SCI-FI']
c = [ 'r','b','y','k']
plt.pie(x,colors=c,labels=l)
plt.show()
```

**OUTPUT :**


**8[b]**

**AIM : Write a program to find kurtosis and skewness of a set of numerical values.**

**INPUT :**

```
# Importing library for kurtosis and skewness
from scipy.stats import kurtosis
from scipy.stats import skew
dataset = [12,13,16,19,20,11,15,16,18,11,12,14,17,20,20]
print("Kurtosis for the given set is : ")
print(kurtosis(dataset, axis=0, bias=True))
print("\nSkewness for the given set is : ")
print(skew(dataset, axis=0, bias=True))
```

**OUTPUT :**

```
Kurtosis for the given set is :
-1.4022409315296627

Skewness for the given set is :
0.0033522818235781605
```

**8[c]**

**AIM : Write a program to find quartile of a set of numerical values at 35 percentile, 60 percentile, and 80 percentile.**

**INPUT :**

```
import numpy as np
marks = [ 85, 56, 64, 88, 78, 76, 75, 72, 96, 100 ]
print("Marks of 10 students are : ", marks )
x = np.percentile(marks, 35)
print("\nQuartile of a set of numerical values at 35 percentile is ",x)
y = np.percentile(marks, 60)
print("\nQuartile of a set of numerical values at 60 percentile is ",y)
z = np.percentile(marks, 80)
print("\nQuartile of a set of numerical values at 80 percentile is ",z)
```

**OUTPUT :**

```
Marks of 10 students are :  [85, 56, 64, 88, 78, 76, 75, 72, 96, 100]

Quartile of a set of numerical values at 35 percentile is  75.15

Quartile of a set of numerical values at 60 percentile is  80.8

Quartile of a set of numerical values at 80 percentile is  89.6
```

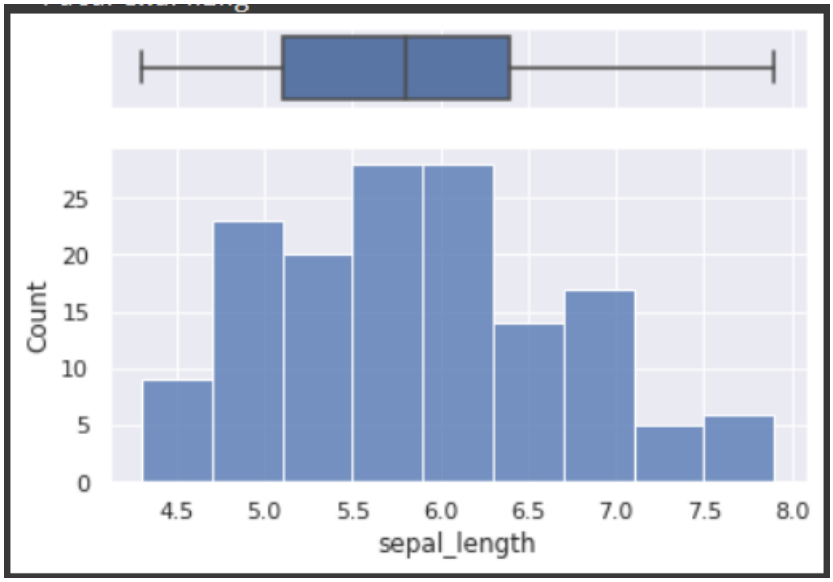
**8[d]**

**AIM : Write a program to demonstrate boxplot and histogram.**

**INPUT :**

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="darkgrid")
df = sns.load_dataset("iris")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"heigh
t_ratios": (.2,.85)})
sns.boxplot(df["sepal_length"], ax=ax_box)
sns.histplot(data=df, x="sepal_length", ax=ax_hist)
ax_box.set(xlabel='')
plt.show()
```

**OUTPUT :**



## PRACTICAL 9

**9[a]**

**AIM : Write a program to drop the columns where at least one element is missing in the given dataframe.**

**INPUT :**

```
import pandas as pd
dt = {"NAME ":["NEEL","PREET","KAVYA","VARSHIT","DEVAM","HARI","HARSH",
"MANTHAN","DHAMO","ARYA"],
      "ROLL-NO.":[28,24,3,9,14,19,16,25,None,6],
      "D.O.B.  ":['28 JAN 2003','24 AUG 2002','03 SEPT 2002','09 MAR 20
03','14 JAN 2001','09 JUL 2003','16 NOV 2003 ','24 SEPT 2003','13 JUL 2
003','21 OCT 2002'],
      "CITY    ":['Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad','Ahemd
abad','Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad','Ahemdabad'],
      "CPI":[9,8,9,8,9,8,8,9,None,9]}
labels=[1,2,3,4,5,6,7,8,9,10]
Student_data = pd.DataFrame(dt,index=labels)
print("Original Data :", "\n\n", Student_data)
Student_data.dropna(axis=1,inplace = True)
print("\n\n New Data after dropping column :", "\n\n", Student_data)
```

**OUTPUT :**

```
Original Data :

       NAME   ROLL-NO.      D.O.B.      CITY     CPI
1      NEEL       28.0   28 JAN 2003  Ahemdabad  9.0
2     PREET       24.0   24 AUG 2002  Ahemdabad  8.0
3     KAVYA        3.0  03 SEPT 2002  Ahemdabad  9.0
4   VARSHIT        9.0   09 MAR 2003  Ahemdabad  8.0
5     DEVAM       14.0   14 JAN 2001  Ahemdabad  9.0
6      HARI       19.0   09 JUL 2003  Ahemdabad  8.0
7     HARSH       16.0  16 NOV 2003   Ahemdabad  8.0
8   MANTHAN       25.0  24 SEPT 2003  Ahemdabad  9.0
9     DHAMO        NaN   13 JUL 2003  Ahemdabad  NaN
10     ARYA        6.0   21 OCT 2002  Ahemdabad  9.0
```

```
 New Data after dropping column :

       NAME        D.O.B.       CITY
1      NEEL    28 JAN 2003  Ahemdabad
2     PREET    24 AUG 2002  Ahemdabad
3     KAVYA   03 SEPT 2002  Ahemdabad
4   VARSHIT    09 MAR 2003  Ahemdabad
5     DEVAM    14 JAN 2001  Ahemdabad
6      HARI    09 JUL 2003  Ahemdabad
7     HARSH   16 NOV 2003   Ahemdabad
8   MANTHAN   24 SEPT 2003  Ahemdabad
9     DHAMO    13 JUL 2003  Ahemdabad
10     ARYA    21 OCT 2002  Ahemdabad
```

**9[b]**

**AIM : Write a program to demonstrate heatmap.**

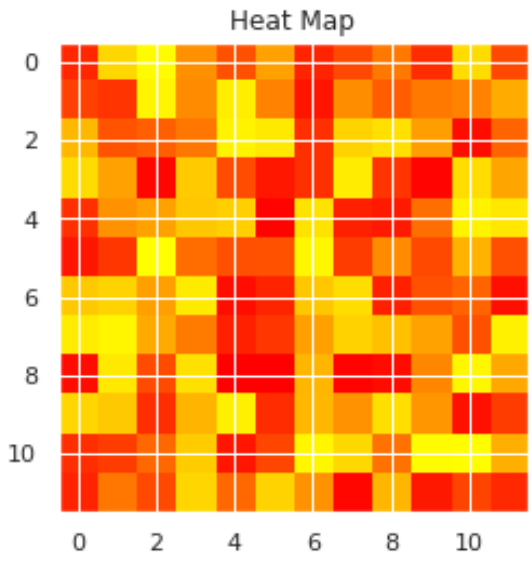
**INPUT :**

```
import numpy as np
import matplotlib.pyplot as plt

data = np.random.random(( 12 , 12 ))
plt.imshow( data , cmap = 'autumn' , interpolation = 'nearest' )

plt.title( "Heat Map" )
plt.show()
```

**OUTPUT :**



**9[c]**

**AIM : Write a program to find mean, median, maximum, minimum, and standard deviation.**

**INPUT :**

```
import numpy as np
list = []
num = int(input("ENTER THE NUMBER OF ELEMENTS YOU WANT IN THE LIST : ")
)

for i in range(1,num+1):
  element = int(input("ENTER THE VALUE OF ELEMENTS : "))
  list.append(element)

print("\nMean = " , np.average(list))
print("Median = " , np.median(list))
print("Max element  = ", np.max(list))
print("Min element = ", np.min(list))
print("Variance = ", np.var(list))
print("Standard Deviation = ", np.std(list))
```

**OUTPUT :**

```
ENTER THE NUMBER OF ELEMENTS YOU WANT IN THE LIST : 5
ENTER THE VALUE OF ELEMENTS : 12
ENTER THE VALUE OF ELEMENTS : 3
ENTER THE VALUE OF ELEMENTS : 78
ENTER THE VALUE OF ELEMENTS : 90
ENTER THE VALUE OF ELEMENTS : 76

Mean =  51.8
Median =  76.0
Max element  =  90
Min element =  3
Variance =  1339.3600000000001
Standard Deviation =  36.59726765757247
```

**9[d]**

**AIM : Write a program to demonstrated undirected graph using NetworkX library.**

**INPUT :**

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
plt.figure(figsize=(6,6))
G.add_edges_from([(1,2),(1,4),(1,6),(2,3),(2,5),(3,5),(4,5),(5,6)])
plt.subplot()
nx.draw_networkx(G)
```

**OUTPUT :**