Experiment No: 1 INTRODUCTION TO NUMPY

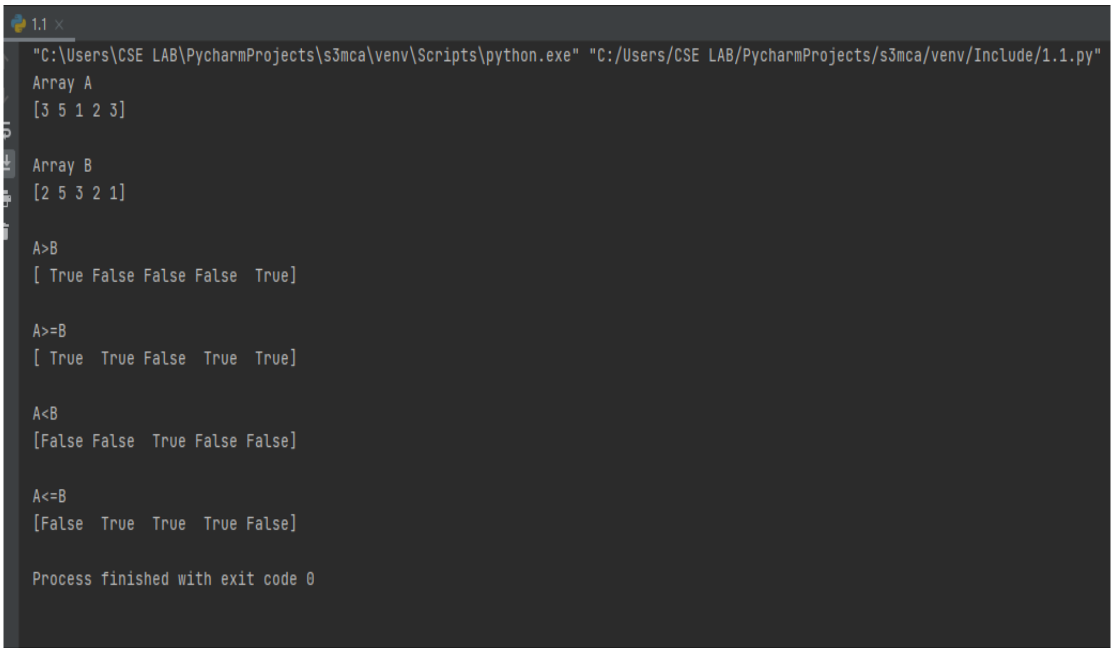
PROGRAM N0-1.1 : ELEMENT WISE COMPARISON OF TWO ARRAYS

AIM : Write a NumPy program to create an element-wise comparison (greater, greater_equal, less and less_equal) of two given arrays

PROGRAM

```
import numpy as np
x = np.array([3,5,1,2,3])
y = np.array([2,5,3,2,1])
print("Array A")
print(x)
print("\nArray B")
print(y)
print("\nA>B")
print(np.greater(x, y))
print("\nA>=B")
print(np.greater_equal(x, y))
print("\nA<B")
print(np.less(x, y))
print("\nA<=B")
print(np.less_equal(x, y))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/Include/1.1.py"
Array A
[3 5 1 2 3]

Array B
[2 5 3 2 1]

A>B
[ True False False False  True]

A>=B
[ True  True False  True  True]

A<B
[False False  True False False]

A<=B
[False  True  True  True False]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

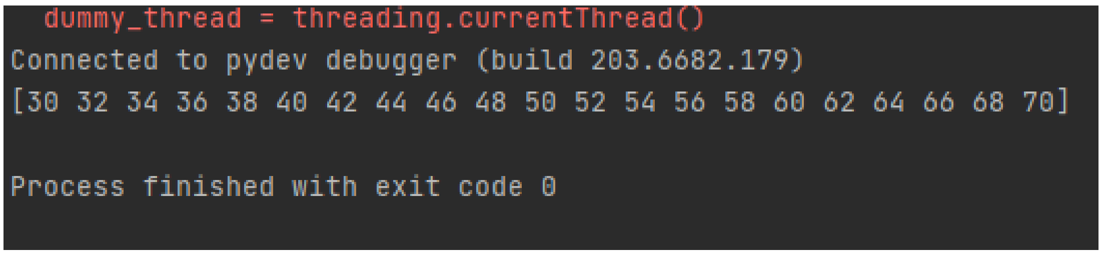
PROGRAM No: 1.2 CREATE AN ARRAY OF ALL EVEN INTEGERS

AIM : Write a NumPy program to create an array of all even integers from 30 to 70

PROGRAM

```
import numpy as np
x = np.arange(start=30, stop=71, step=2)
print(x)
```

OUTPUT



```
  dummy_thread = threading.currentThread()
Connected to pydev debugger (build 203.6682.179)
[30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

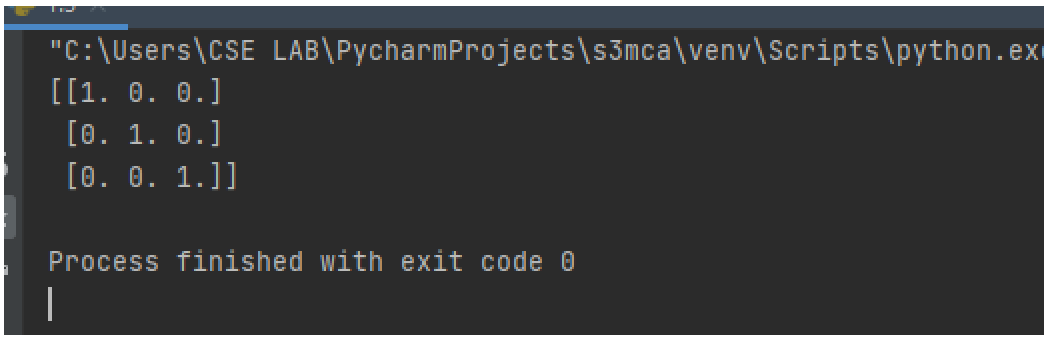
PROGRAM No: 1. 3 CREATE A 3X3 IDENTITY MATRIX

AIM : Write a NumPy program to create a 3x3 identity matrix.

PROGRAM

```
import numpy as np

x = np.identity(3)
print(x)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.ex
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM No: 1. 4

CREATE A VECTOR WITH VALUES FROM 0 TO 20 AND CHANGE THE SIGN OF THE NUMBERS IN RANGE FROM 9 TO 15

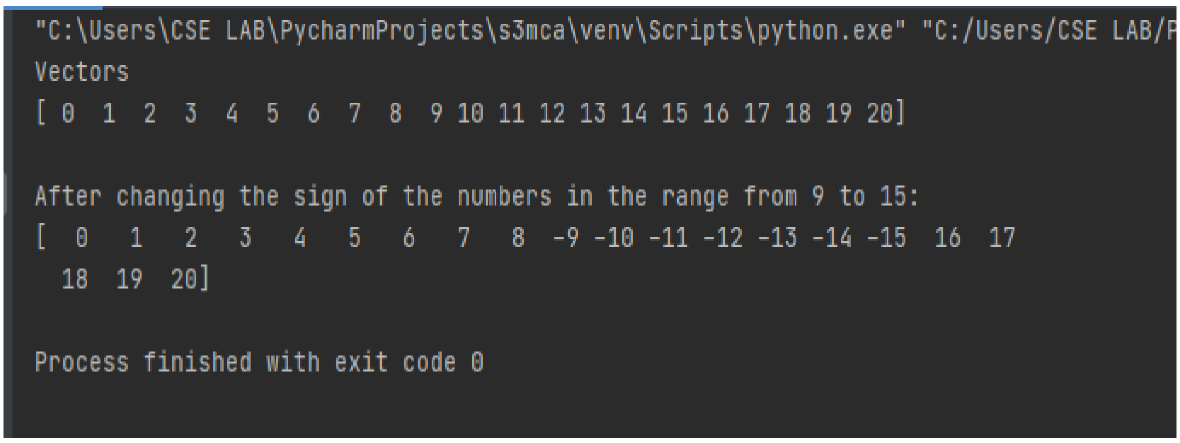
AIM : Write a NumPy program to create a vector with values from 0 to 20 and change the sign of

the numbers in the range from 9 to 15.

PROGRAM

```
import numpy as np

x = np.arange(21)

print("Vectors ")

print(x)

print("\nAfter changing the sign of the numbers in the range from 9 to 15:")

x[(x >= 9) & (x <= 15)] *= -1

print(x)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/P
Vectors
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]

After changing the sign of the numbers in the range from 9 to 15:
[  0   1   2   3   4   5   6   7   8  -9 -10 -11 -12 -13 -14 -15  16  17
  18  19  20]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM No: 1.5

CREATE A 5X5 ZERO MATRIX WITH ELEMENTS ON THE MAIN DIAGONAL EQUAL TO 1,2,3,4,5

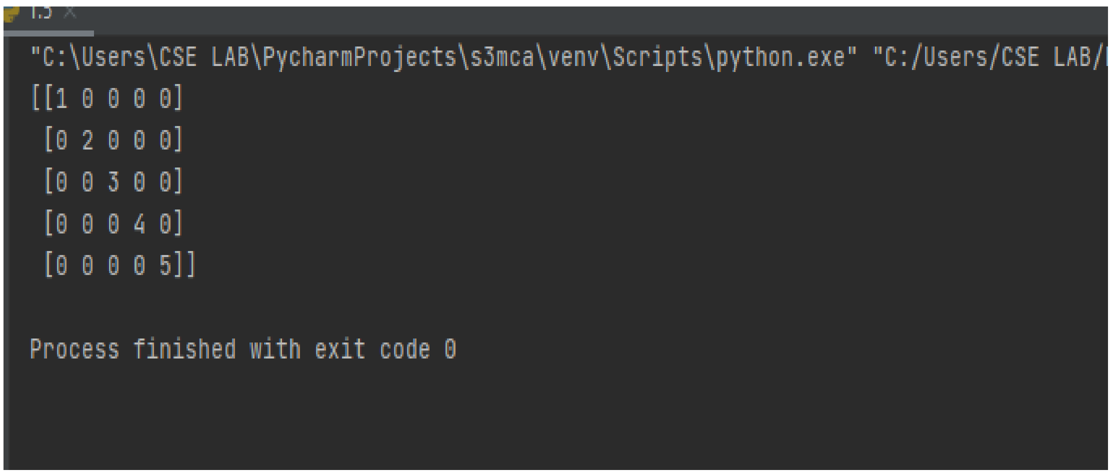
AIM : Write a NumPy program to create a 5x5 zero matrix with elements on the main diagonal

equal to 1, 2, 3, 4, 5.

PROGRAM

```
import numpy as np
x = np.diag([1, 2, 3, 4, 5])
print(x)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/
[[1 0 0 0 0]
 [0 2 0 0 0]
 [0 0 3 0 0]
 [0 0 0 4 0]
 [0 0 0 0 5]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

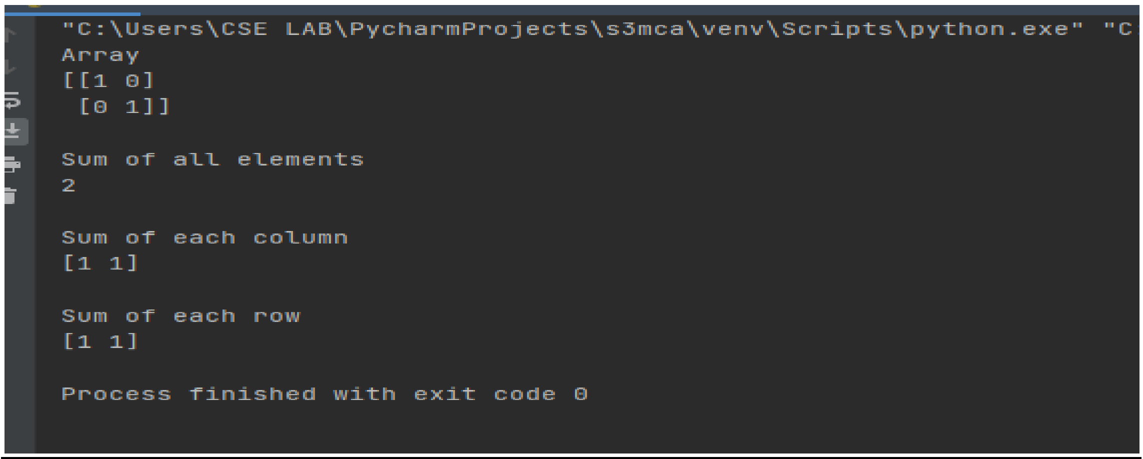
PROGRAM No: 1.6 SUM OF ALL ELEMENTS OF A GIVEN ARRAY

AIM : Write a NumPy program to compute sum of all elements, sum of each column and sum of each row of a given array.

PROGRAM

```
import numpy as np

x = np.array([[1,0],[0,1]])

print("Array")

print(x)

print("\nSum of all elements")

print(np.sum(x))

print("\nSum of each column")

print(np.sum(x, axis=0))

print("\nSum of each row")

print(np.sum(x, axis=1))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C
Array
[[1 0]
 [0 1]]

Sum of all elements
2

Sum of each column
[1 1]

Sum of each row
[1 1]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM No: 1. 7

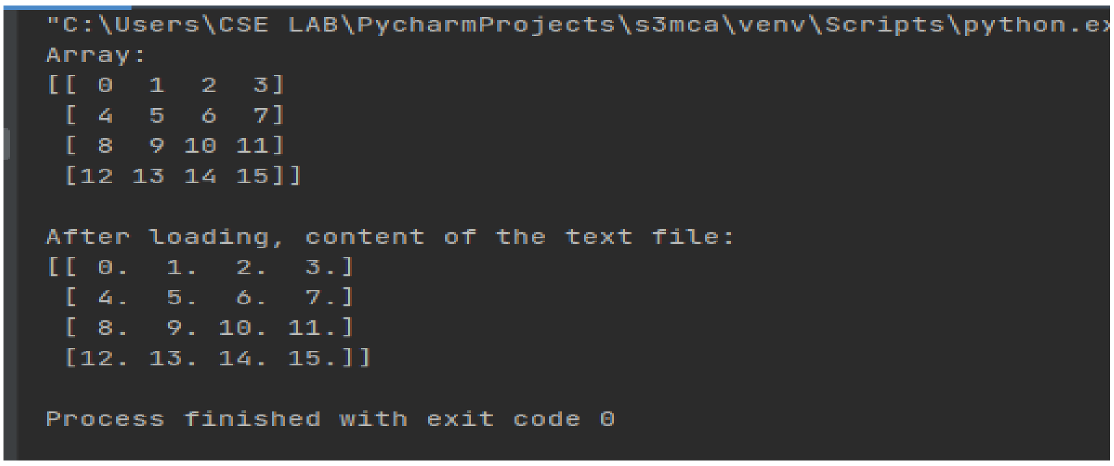
SAVE A GIVE ARRAY TO A TEXT FILE AND LOAD IT

AIM : Write a NumPy program to save a given array to a text file and load it.

PROGRAM

```
import numpy as np

import os

x = np.arange(16).reshape(4,4)

print("Array:")

print(x)

header = 'C1 C2 C3 C4'

np.savetxt('Array.txt', x, fmt="%d", header=header)

print("\nAfter loading, content of the text file:")

print(np.loadtxt('Array.txt'))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.ex
Array:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]

After loading, content of the text file:
[[ 0.  1.  2.  3.]
 [ 4.  5.  6.  7.]
 [ 8.  9. 10. 11.]
 [12. 13. 14. 15.]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM No: 1. 8

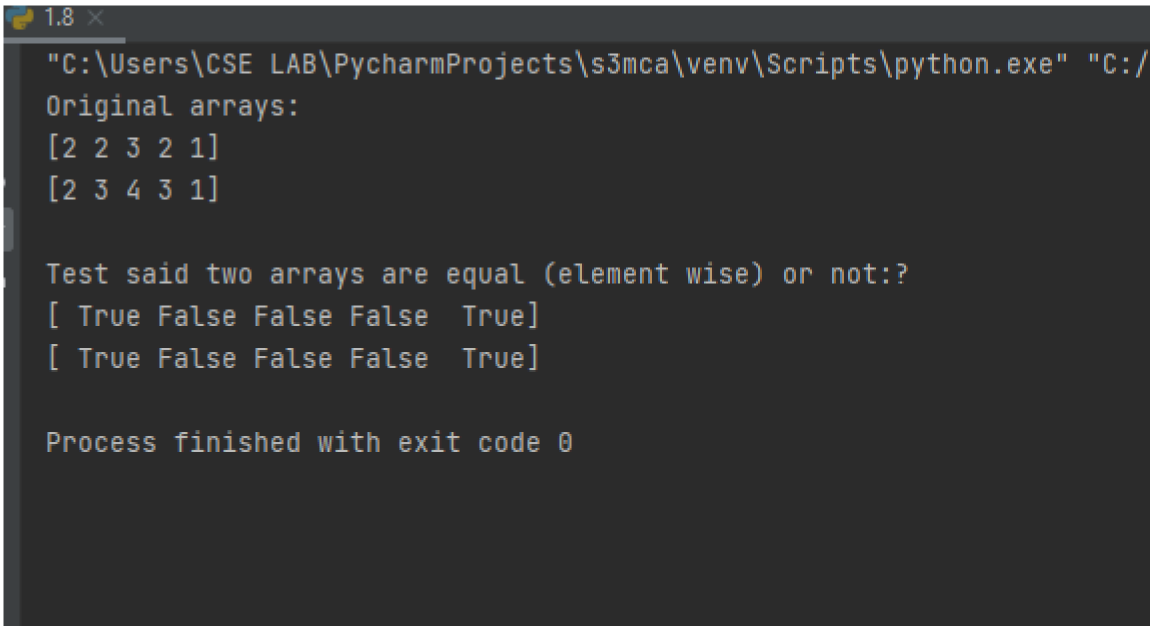
CHECK WHETHER TWO ARRAYS ARE EQUAL OR NOT

AIM : Write a NumPy program to check whether two arrays are equal (element wise) or not

PROGRAM

```
import numpy as np
nums1 = np.array([2,2,3,2,1])
nums2 = np.array([2,3,4,3,1])
print("Original arrays:")
print(nums1)
print(nums2)
print("\nTest said two arrays are equal (element wise) or not:?")
print(nums1 == nums2)
print(np.equal(nums1, nums2))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/
Original arrays:
[2 2 3 2 1]
[2 3 4 3 1]

Test said two arrays are equal (element wise) or not:?
[ True False False False  True]
[ True False False False  True]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM NO - 1.9

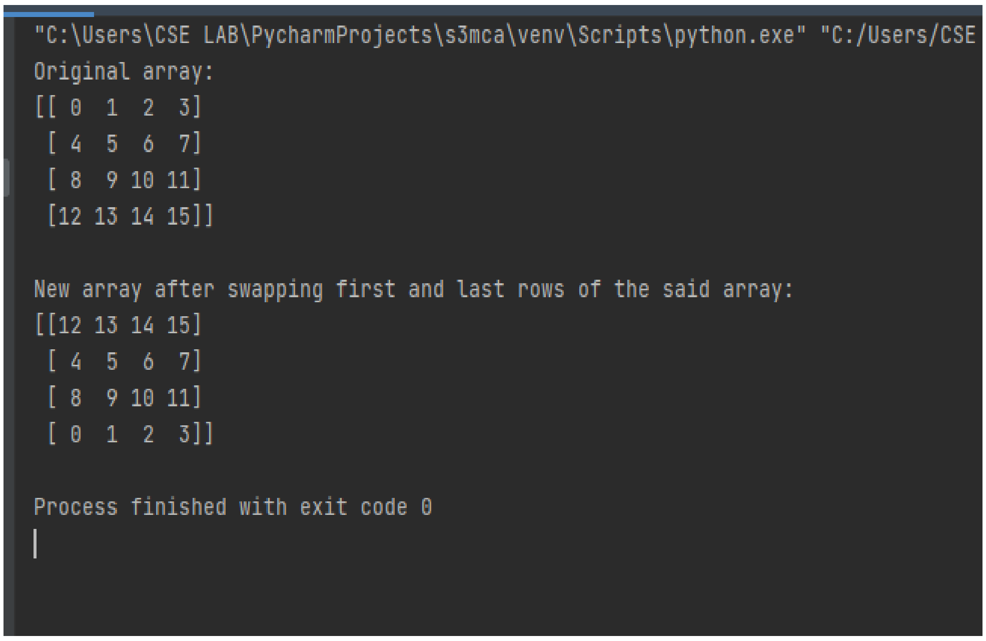
CREATE A 4X4 ARRAY WITH RANDOM VALUES AND SWAPPING FIRST AND LAST ROWS.

AIM : Write a NumPy program to create a 4x4 array with random values, now create a new array from the said array swapping first and last rows.

PROGRAM

```
import numpy as np
nums = np.arange(16, dtype='int').reshape(-1, 4)
print("Original array:")
print(nums)
print("\nNew array after swapping first and last rows of the said array:")
nums = nums[[-1,1,2,0]]
print(nums)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE
Original array:
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]

New array after swapping first and last rows of the said array:
[[12 13 14 15]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [ 0  1  2  3]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM No: 1. 10

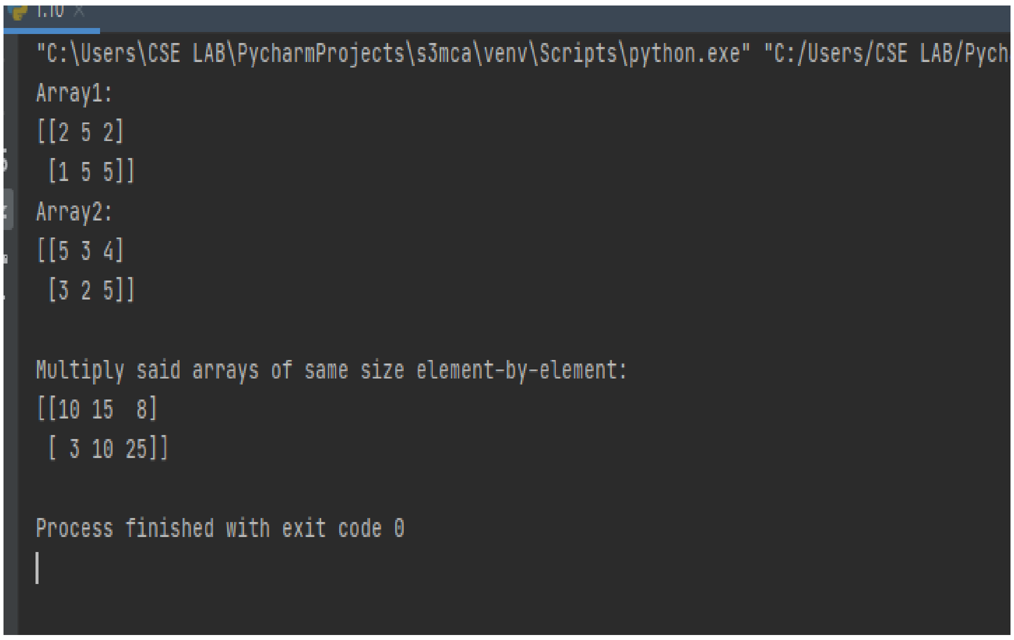
MULTIPLY TWO GIVEN ARRAYS OF SAME SIZE ELEMENT-BY-ELEMENT.

AIM : Write a NumPy program to multiply two given arrays of same size element-by-element.

PROGRAM

```
import numpy as np
nums1 = np.array([[2, 5, 2],[1, 5, 5]])
nums2 = np.array([[5, 3, 4],[3, 2, 5]])
print("Array1:")
print(nums1)
print("Array2:")
print(nums2)
print("\nMultiply said arrays of same size element-by-element:")
print(np.multiply(nums1, nums2))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/Pych
Array1:
[[2 5 2]
 [1 5 5]]
Array2:
[[5 3 4]
 [3 2 5]]

Multiply said arrays of same size element-by-element:
[[10 15  8]
 [ 3 10 25]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

Experiment No: 2

MATRIX OPERATIONS

AIM : Write Python program to create two matrices (read values from user) and find the following

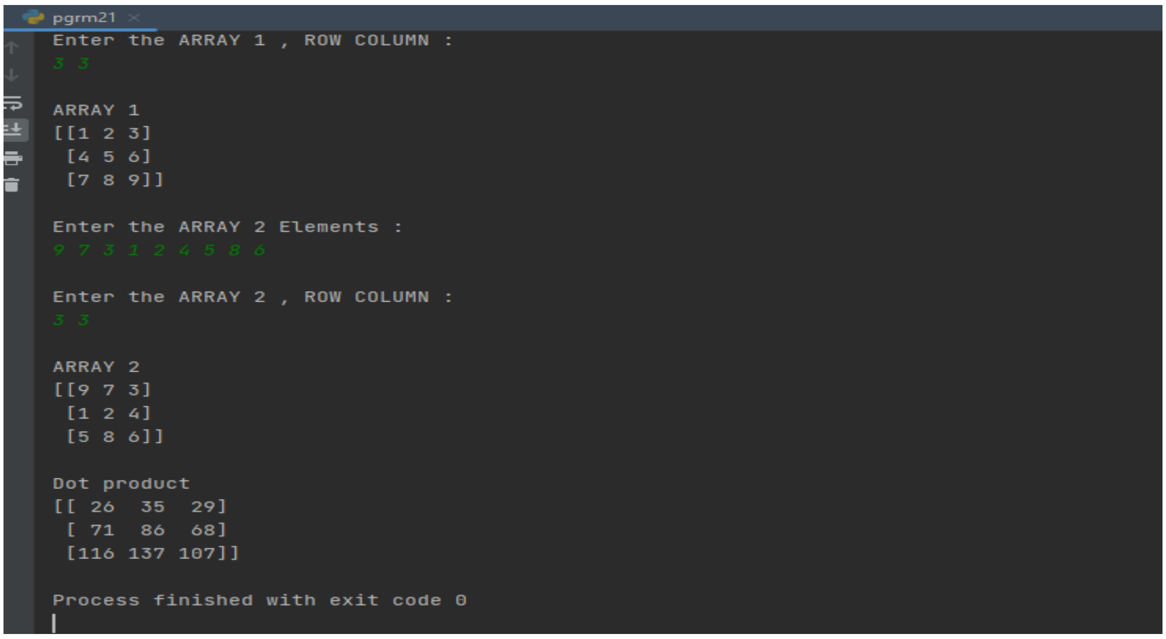
1. Dot Product
2. Transpose
3. Trace
4. Rank
5. Determinant
6. Inverse
7. Eigen values and eigen vectors

PROGRAM

1) Dot Product

```
import numpy as np
def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))
    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):
        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    return array_1

arr1 = create_matrix(1)
arr2 = create_matrix(2)
if(arr1.shape == arr2.shape):
    print("\nDot product")
    print(np.dot(arr1,arr2))
else:
    print("\nDimensions not matching!")
```

OUTPUT

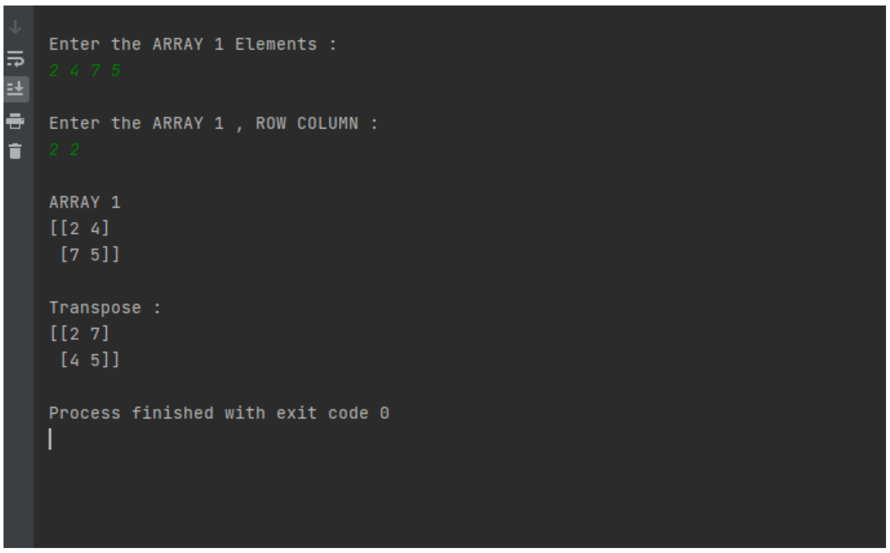


```
Enter the ARRAY 1 , ROW COLUMN :
3 3

ARRAY 1
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Enter the ARRAY 2 Elements :
9 7 3 1 2 4 5 8 6

Enter the ARRAY 2 , ROW COLUMN :
3 3

ARRAY 2
[[9 7 3]
 [1 2 4]
 [5 8 6]]

Dot product
[[ 26  35  29]
 [ 71  86  68]
 [116 137 107]]

Process finished with exit code 0
```

2) TRANSPOSE

```
import numpy as np
def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))

    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):

        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    print("\nTranspose : ")
    return array_1
    print(create_matrix(1)
    .transpose())
```

OUTPUT:

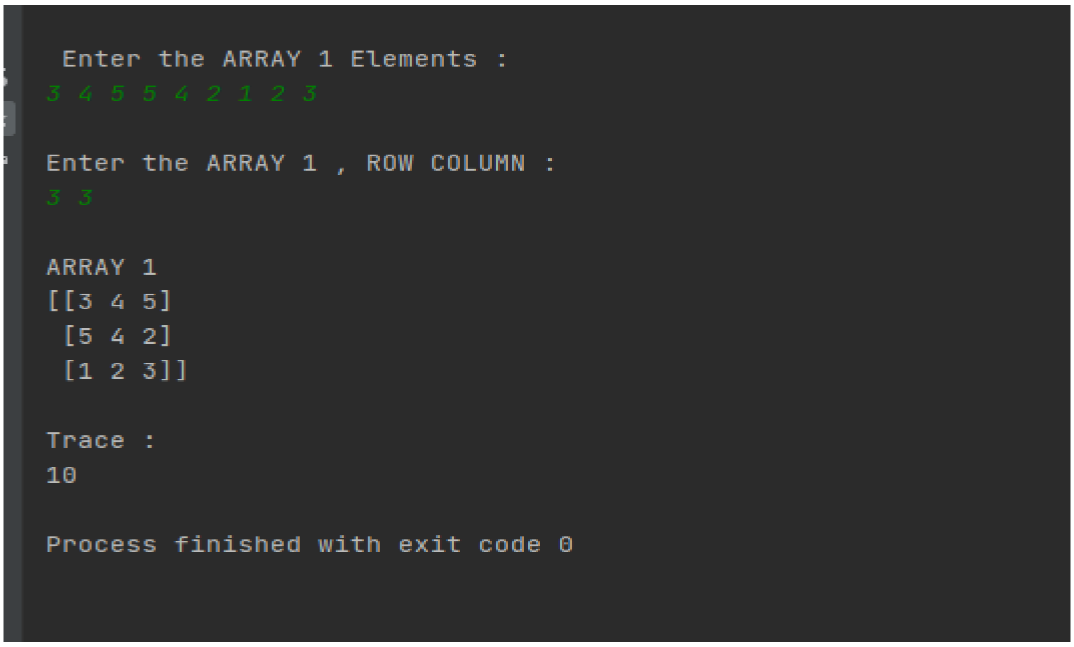


```
Enter the ARRAY 1 Elements :
2 4 7 5

Enter the ARRAY 1 , ROW COLUMN :
2 2

ARRAY 1
[[2 4]
 [7 5]]

Transpose :
[[2 7]
 [4 5]]

Process finished with exit code 0
```

<u>3) TRACE</u>

```
import numpy as np
def create_matrix(mc):
    print("\n Enter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))
    #print(arr)
    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):
        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    print("\nTrace : ")
    return array_1
print(create_matrix(1).trace())
```

OUTPUT:

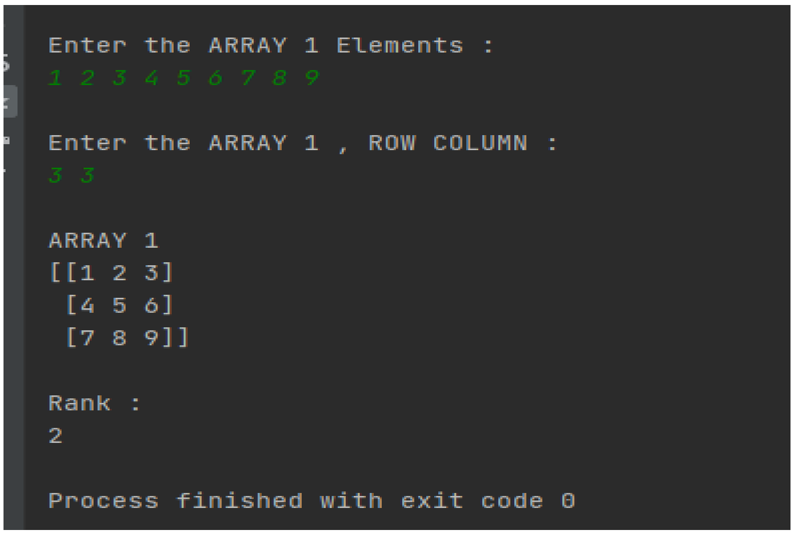


```
 Enter the ARRAY 1 Elements :
3 4 5 5 4 2 1 2 3

Enter the ARRAY 1 , ROW COLUMN :
3 3

ARRAY 1
[[3 4 5]
 [5 4 2]
 [1 2 3]]

Trace :
10

Process finished with exit code 0
```

4) RANK

```
import numpy as np

def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))
    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):
        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    print("\nRank : ")
    return array_1
print(np.linalg.matrix_rank(create_matrix(1)))
```

OUTPUT:

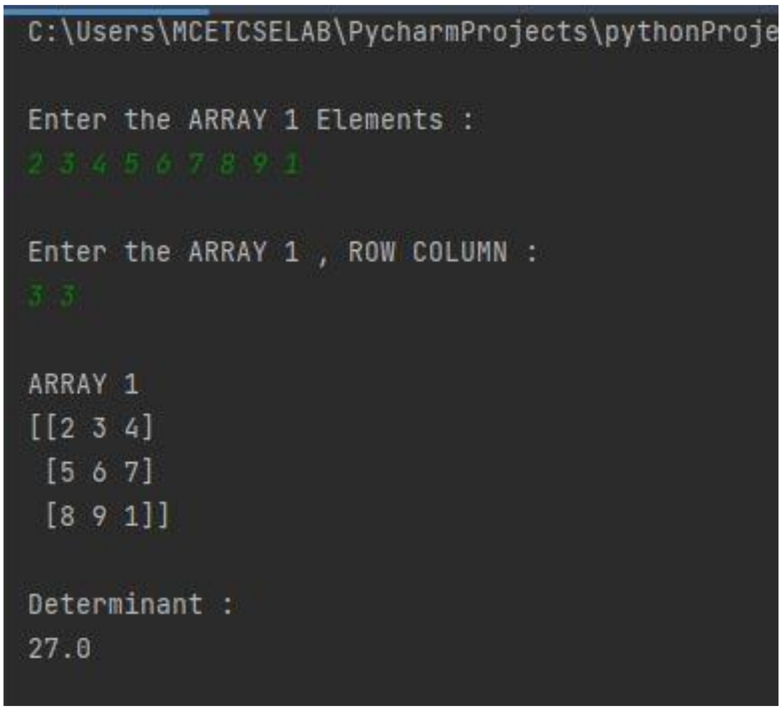


```
Enter the ARRAY 1 Elements :
1 2 3 4 5 6 7 8 9

Enter the ARRAY 1 , ROW COLUMN :
3 3

ARRAY 1
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Rank :
2

Process finished with exit code 0
```

5) DETERMINANT

```
import numpy as np
def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))
    #print(arr)
    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):
        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    print("\nDeterminant : ")
    return array_1
print(np.linalg.det(create_matrix(1)))
```

OUTPUT:

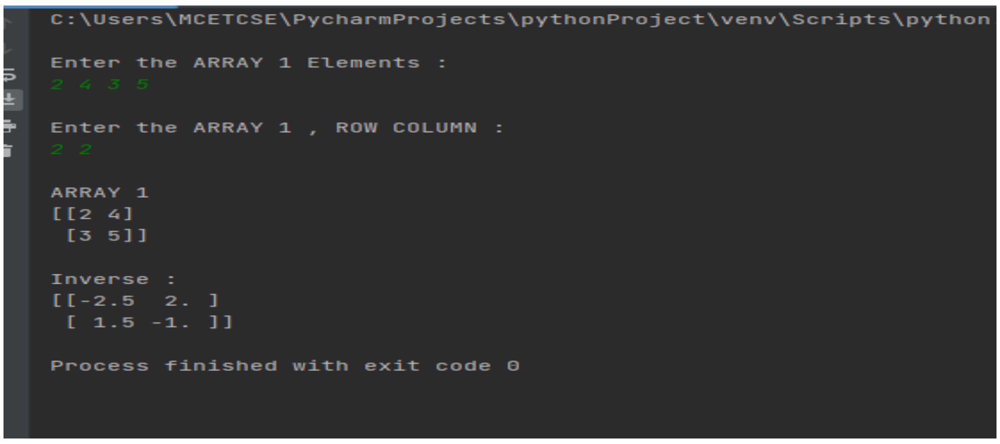


```
C:\Users\MCETCSELAB\PycharmProjects\pythonProje

Enter the ARRAY 1 Elements :
2 3 4 5 6 7 8 9 1

Enter the ARRAY 1 , ROW COLUMN :
3 3

ARRAY 1
[[2 3 4]
 [5 6 7]
 [8 9 1]]

Determinant :
27.0
```

<u>6) INVERSE</u>

```
import numpy as np
def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))

    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")

    row,column = map(int, input().split())

    if(len(array_1)!= (row*column)):

        print("\nRow and Column size not match with total elements !! retry")

        return create_matrix(mc)

    array_1 = array_1.reshape(row,column)

    print("\nARRAY "+str(mc))

    print(array_1)

    print("\nInverse : ")

    return array_1

print(np.linalg.inv(create_matrix(1)))
```

OUTPUT:

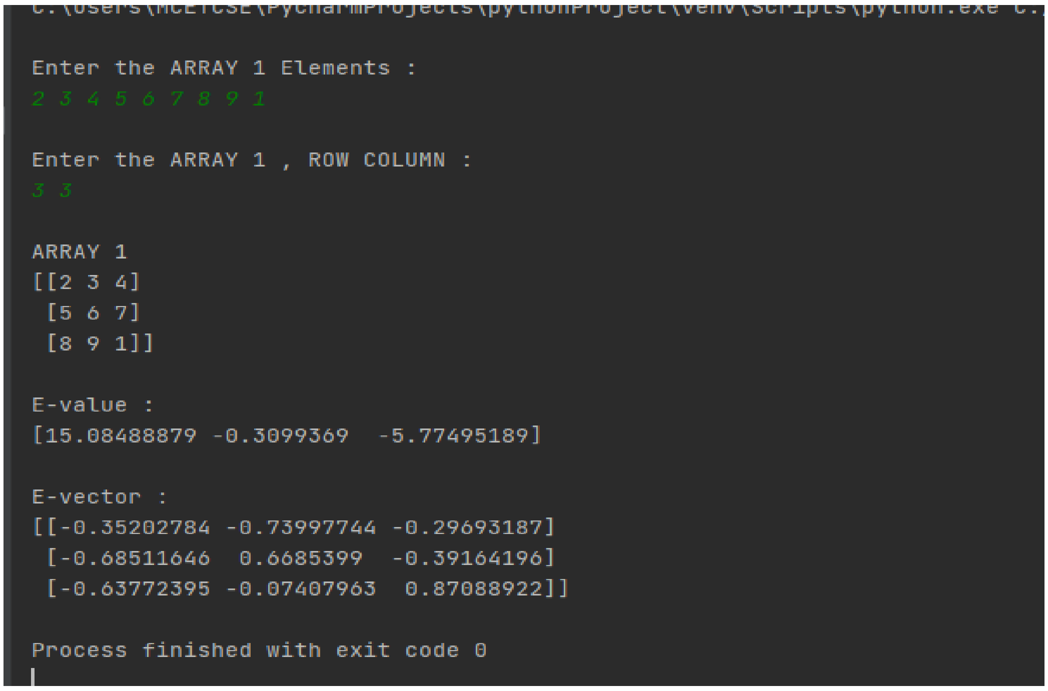


```
C:\Users\MCETCSE\PycharmProjects\pythonProject\venv\Scripts\python

Enter the ARRAY 1 Elements :
2 4 3 5

Enter the ARRAY 1 , ROW COLUMN :
2 2

ARRAY 1
[[2 4]
 [3 5]]

Inverse :
[[-2.5  2. ]
 [ 1.5 -1. ]]

Process finished with exit code 0
```

7) EIGEN VALUES AND EIGEN VECTORS

```
import numpy as np
def create_matrix(mc):
    print("\nEnter the ARRAY "+str(mc)+" Elements : ")
    array_1 = map(int, input().split())
    array_1 = np.array(list(array_1))
    #print(arr)
    print("\nEnter the ARRAY "+str(mc)+" , ROW COLUMN : ")
    row,column = map(int, input().split())
    if(len(array_1)!= (row*column)):
        print("\nRow and Column size not match with total elements !! retry")
        return create_matrix(mc)
    array_1 = array_1.reshape(row,column)
    print("\nARRAY "+str(mc))
    print(array_1)
    return array_1
x,y = np.linalg.eig(create_matrix(1))
print("\nE-value : ")
print(x)
print("\nE-vector : ")

print(y)
```

OUTPUT:



```
C:\Users\MCETCSE\PycharmProjects\pythonProject\venv\Scripts\python.exe C:

Enter the ARRAY 1 Elements :
2 3 4 5 6 7 8 9 1

Enter the ARRAY 1 , ROW COLUMN :
3 3

ARRAY 1
[[2 3 4]
 [5 6 7]
 [8 9 1]]

E-value :
[15.08488879 -0.3099369  -5.77495189]

E-vector :
[[-0.35202784 -0.73997744 -0.29693187]
 [-0.68511646  0.6685399  -0.39164196]
 [-0.63772395 -0.07407963  0.87088922]]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

Experiment No: 3

PROGRAMS USING MATPLOTLIB, PLOTLY AND SEABORN LIBRARY

PROGRAM NO- 3.1

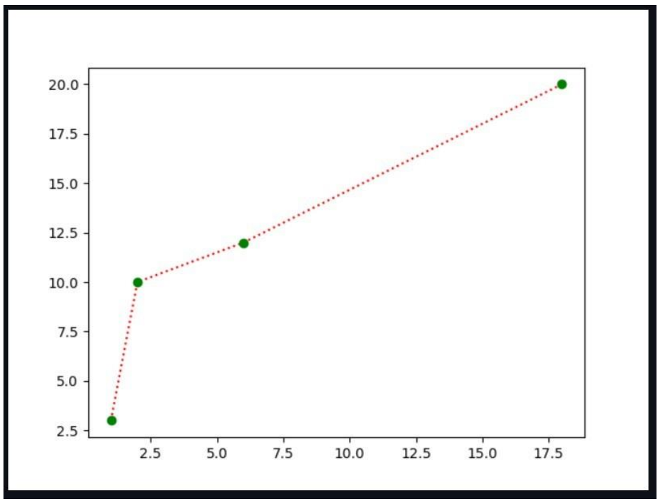
DRAW A LINE FROM A GIVEN POINT

AIM : . Draw a line in a diagram from position (1, 3) to (2, 10) then to (6, 12) and finally to position (18, 20). (Mark each point with a beautiful green colour and set line colour to red and line style dotted)

PROGRAM

```
import matplotlib.pyplot as plt
import numpy as np
xpoints = np.array([1, 2, 6, 18])
ypoints = np.array([3, 10, 12, 20])
plt.plot(xpoints, ypoints,marker = 'o',color="red",mec = 'g', mfc = 'g',linestyle = 'dotted')
plt.show()
```

OUTPUT



RESULT

The program has been executed and verified successfully.

PROGRAM No: 3.2

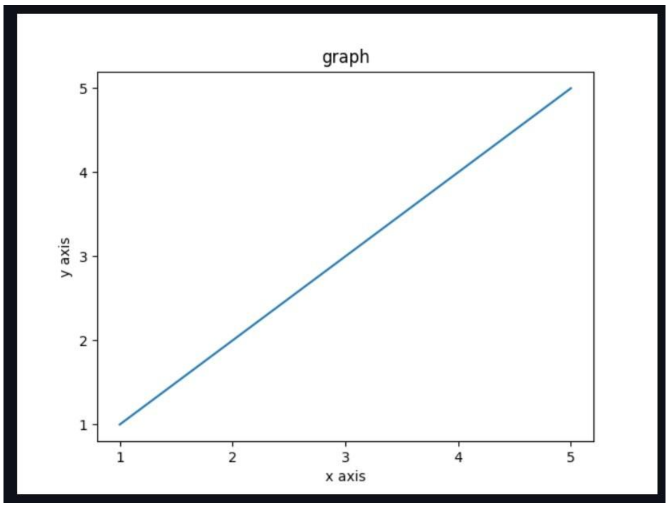
DRAW ALINE USING VALUES TAKEN FROM A TEXT FILE AND LABEL IT

AIM: Write a Python program to draw a line using given axis values taken from a text file, with suitable label in the x axis, y axis and a title

PROGRAM

```
import matplotlib.pyplot as plt
with open("3_data.txt") as f:
data = f.read()
data = data.split('\n')
x = [row.split(' ')[0] for row in data]
y = [row.split(' ')[1] for row in data]
plt.plot(x, y)
plt.xlabel('x axis')
plt.ylabel('y axis')
plt.title('graph')
plt.show()
```

OUTPUT



RESULT

The program has been executed and verified successfully.

PROGRAM No: 3.3

IRIS DATASET VISUALIZATION USING PYPLOTAND SEABORN LIBRARY

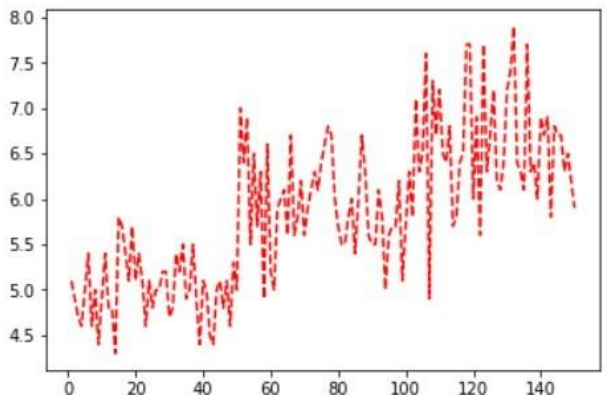
AIM : Write apython to visualize the iris dataset

PROGRAM

(a) Plotting speal length and petal length

```
import pandas as pd

import matplotlib.pyplot as plt

iris = pd.read_csv("Iris.csv")

plt.plot(iris.Id, iris["SepalLengthCm"], "r--")

plt.show()
```

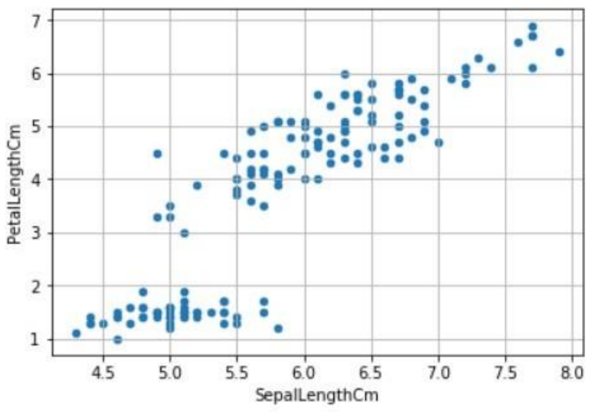
OUTPUT



(b) Scatterplot

```
    iris.plot(kind ="scatter",

    x ='SepalLengthCm',

    y ='PetalLengthCm')

    plt.grid()
```

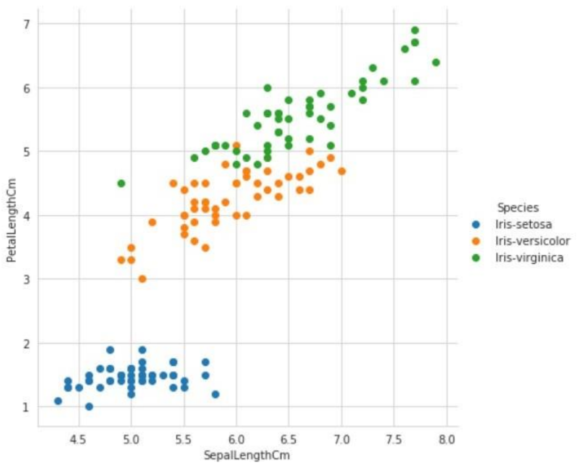
OUTPUT



(c) **Plot using seaborn library**

```
import seaborn as sns

iris = sns.load_dataset('iris')

sns.set_style("whitegrid")

sns.FacetGrid(iris, hue ="species",height = 6).map(plt.scatter,
'sepal_length','petal_length').add_legend()
```

OUTPUT



RESULT

The program has been executed and verified successfully.

PROGRAM No: 3.4

CREATE PIE CHART

AIM : Write a Python programming to create a pie chart of gold medal achievements of five most

successful countries in 2016 Summer Olympics. Read the data from a csv file.

Sample data:

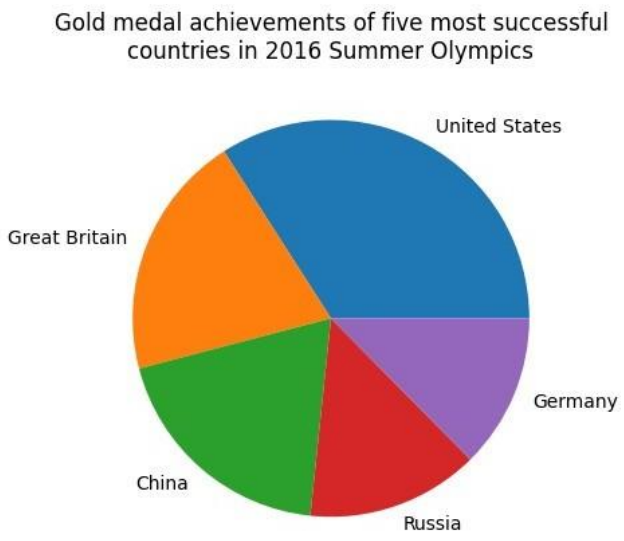
medal.csv

```
country,gold_medal

United States,46

Great Britain,27

China,26

Russia,19

Germany,17
```

PROGRAM

```
import matplotlib.pyplot as plt

import pandas as pd

df = pd.read_csv('9_data.csv')

country_data =  df["country"]

medal_data = df["gold_medal"]

plt.pie(medal_data, labels=country_data)

plt.title("Gold medal achievements of five most successful\n"+"countries in 2016 Summer Olympics")

plt.show()
```

OUTPUT



RESULT

The program has been executed and verified successfully.

PROGRAM No: 3.6

CREATE BAR CHART

AIM : Consider the following data.

Programming languages: Java Python PHP JavaScript C# C++

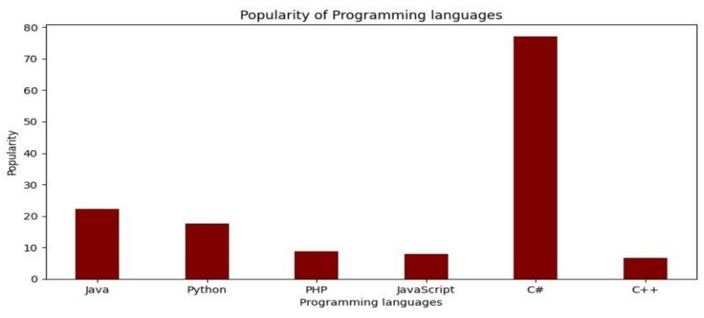
Popularity: 22.2 17.6 8.8 8 7.7 6.7

Write a Python programming to display a bar chart of the popularity of programming Languages.

PROGRAM

```
import numpy as np
import matplotlib.pyplot as plt
data = {'Java':22.2, 'Python':17.6,
'PHP':8.8,'JavaScript':8,'C#':7.7,'C++':6.7}
courses = list(data.keys())
values = list(data.values())
fig = plt.figure(figsize = (10, 5))
plt.bar(courses, values, color ='maroon',width = 0.4)
plt.xlabel("Programming languages")
plt.ylabel("Popularity")
plt.title("Popularity of Programming languages")
plt.show()
```

OUTPUT



RESULT

The program has been executed and verified successfully.

Experiment No: 4 INTRODUCTION TO PANDAS

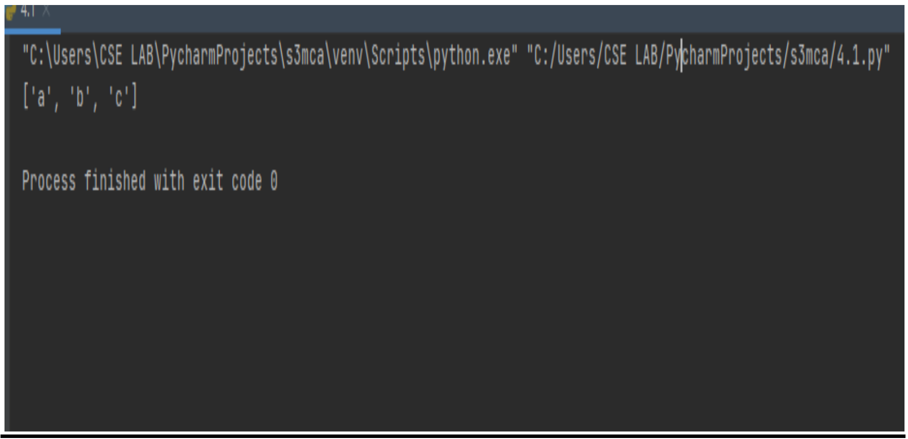
PROGRAM N0-4.1 List-to-Series Conversion

AIM : Write a python program to implement List-to-Series Conversion

PROGRAM

```
import pandas as pd

names = ['a','b','c']

x = pd.Series(names)

print(names)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/4.1.py"
['a', 'b', 'c']

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

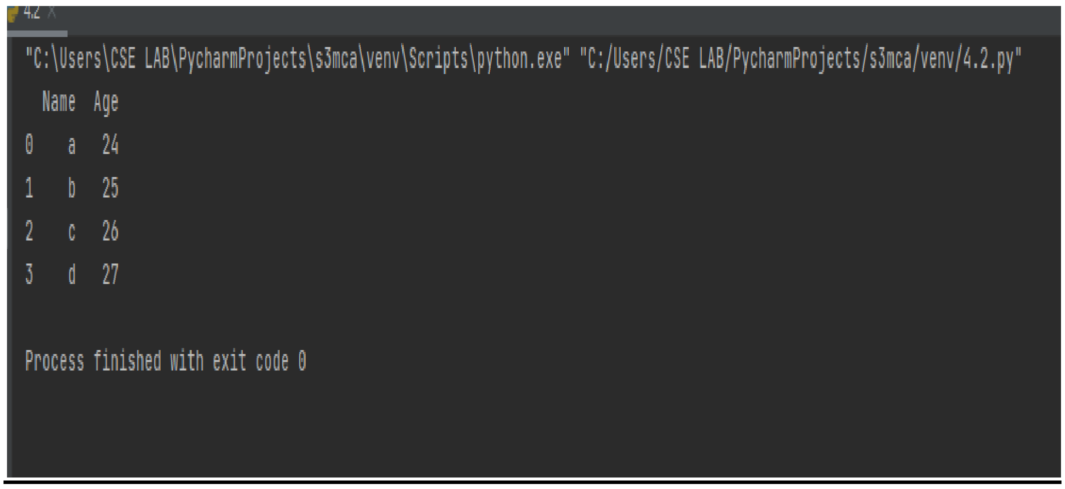
PROGRAM N0-4.2 DICTIONARY TO DATAFRAME CONVERSION

AIM : Write a python program to convert the given a dictionary into corresponding dataframe and display it.

PROGRAM

```
import pandas as pd

details = {'Name' : ['a','b','c','d'],'Age' : [24,25,26,27],}

df = pd.DataFrame(details)

print(df)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/4.2.py"
  Name  Age
0    a   24
1    b   25
2    c   26
3    d   27

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

PROGRAM N0-4.3 CSV FILE TO DATAFRAME CONVERSION

AIM : Write a python program to read the given CSV file, and convert it into a dataframe and display it.

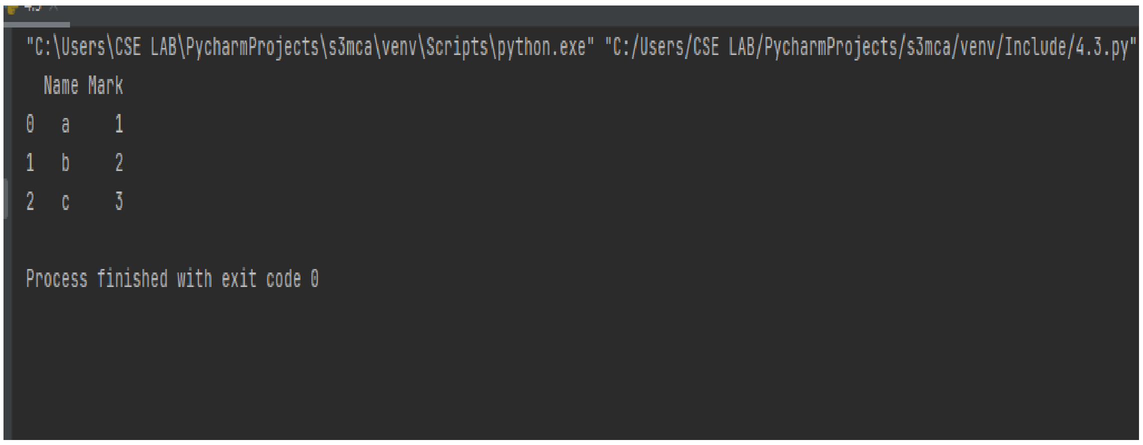
PROGRAM

```
import pandas as pd

df = pd.read_csv('file1.csv')

print(df.to_string())
```

file1.csv

Name	mark
a	1
b	2
c	3

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/Include/4.3.py"
  Name Mark
0   a    1
1   b    2
2   c    3

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

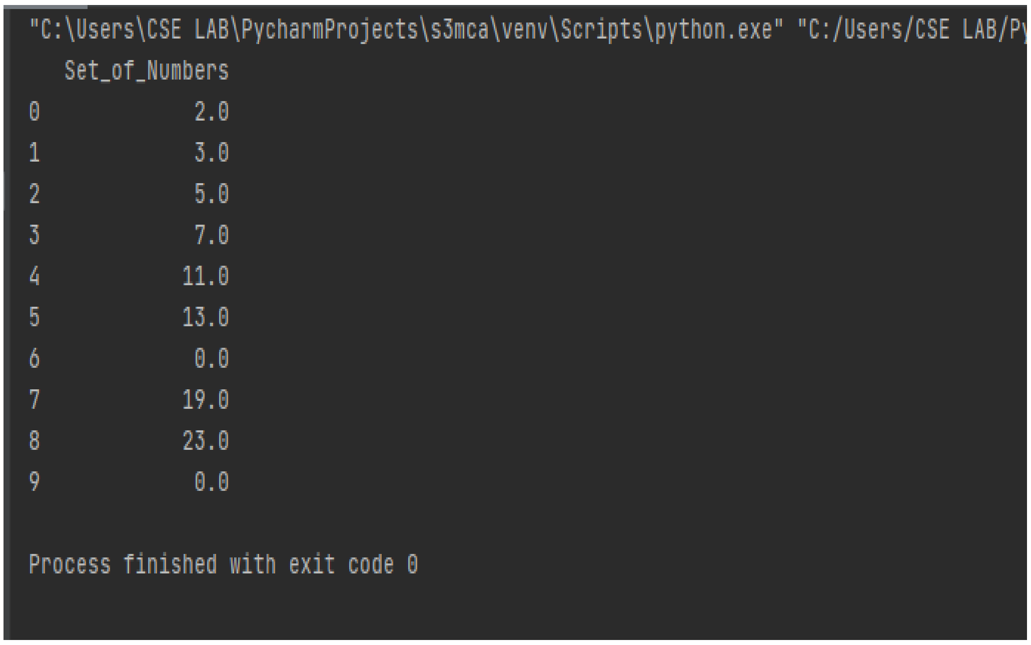
PROGRAM N0-4.4 FILL THE NAN VALUES WITH 0 FOR A GIVEN A DATAFRAME

AIM : Write a python program to fill the NaN values with 0 for a given a dataframe with NaN Values.

PROGRAM

```
import pandas as pd
import numpy as np
nums = {'Set_of_Numbers': [2, 3, 5, 7, 11, 13,np.nan, 19, 23, np.nan]}
df = pd.DataFrame(nums, columns =['Set_of_Numbers'])
df['Set_of_Numbers'] = df['Set_of_Numbers'].fillna(0)
print(df)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/Py
   Set_of_Numbers
0             2.0
1             3.0
2             5.0
3             7.0
4            11.0
5            13.0
6             0.0
7            19.0
8            23.0
9             0.0

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

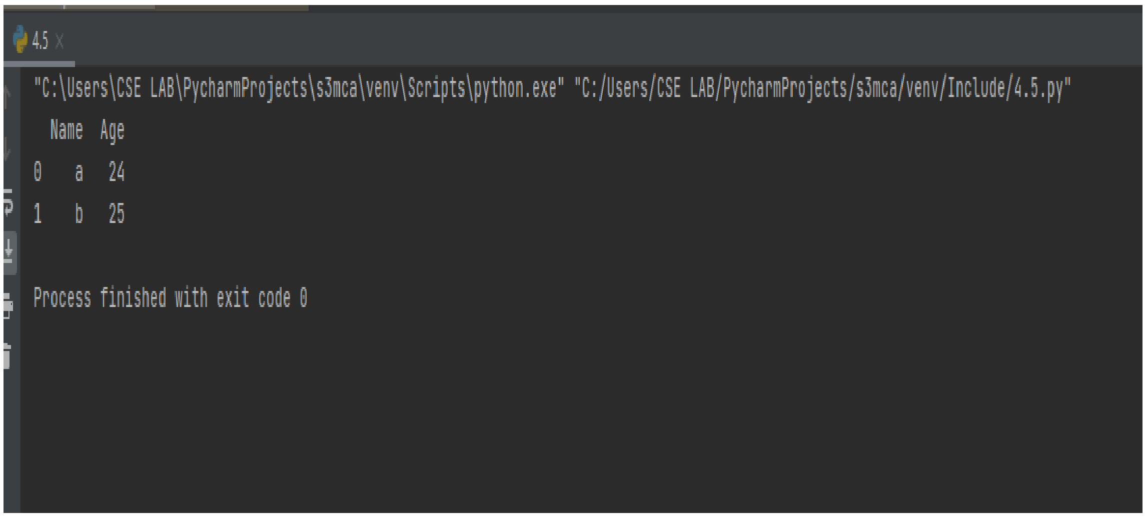
PROGRAM N0-4.5 SELECT FIRST 2 ROWS AND OUTPUT THEM FROM A GIVEN A DATAFRAME.

AIM : Write a python program to select first 2 rows and output them from a given a dataframe.

PROGRAM

```
import pandas as pd
details = {'Name' : ['a','b','c','d'],'Age' : [24,25,26,27],}
df = pd.DataFrame(details)
print(df[:2])
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/Include/4.5.py"
  Name  Age
0    a   24
1    b   25


Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

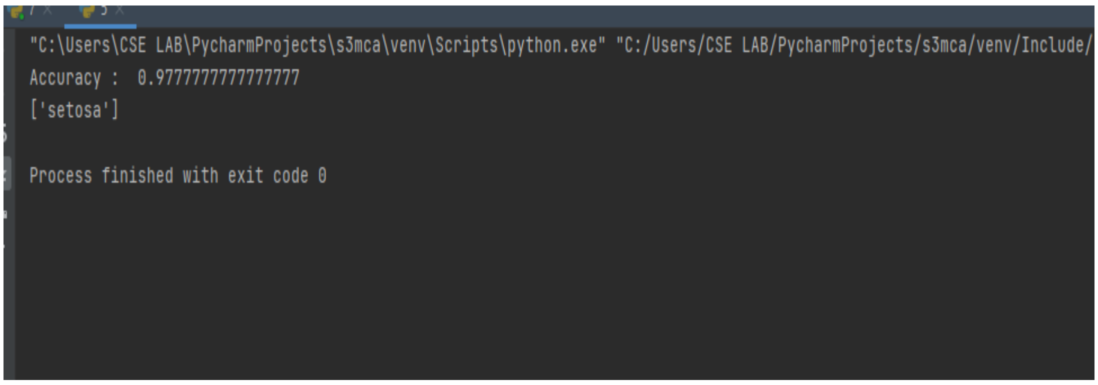
Experiment No: 5 K-NN CLASSIFICATION ALGORITHM

AIM : Write a python program to implement K-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm.

PROGRAM

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
rom sklearn import metrics
iris = load_iris()
x = iris.data
y = iris.target
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=1)
c_knn = KNeighborsClassifier(n_neighbors=3)
c_knn.fit(x_train,y_train)
y_pred = c_knn.predict(x_test)
print("Accuracy : ",metrics.accuracy_score(y_test,y_pred))
sample = [[2,2,2,2]]
pred = c_knn.predict(sample)
pred_v = [iris.target_names[p] for p in pred]
print(pred_v)
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/Include/
Accuracy :  0.9777777777777777
['setosa']

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully

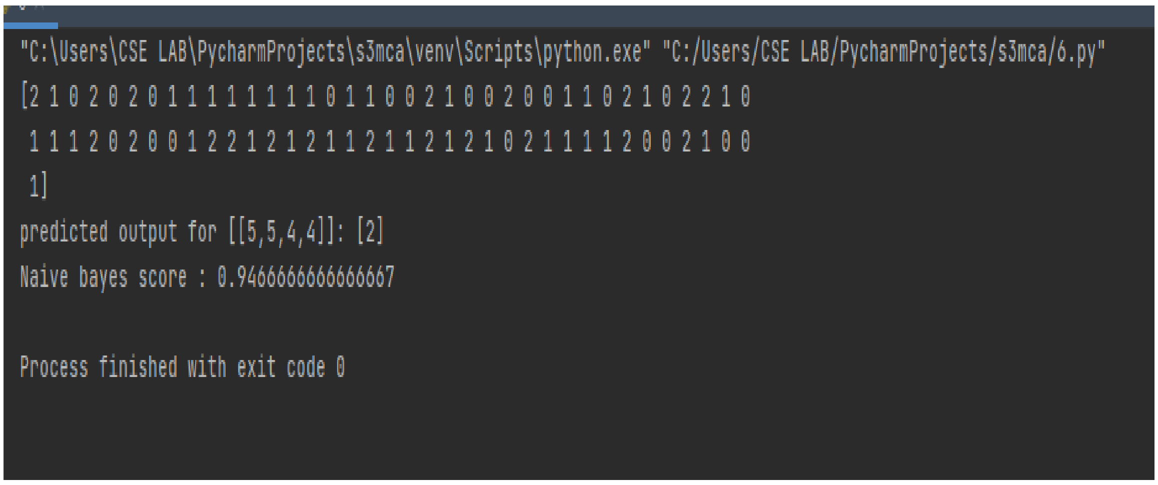
Experiment No: 6 NAIVE BAYES CLASSIFICATION ALGORITHM

AIM : Write a python program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm.

PROGRAM

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
X,y=load_iris(return_X_y=True)
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.5,random_state=0)
gnb=GaussianNB()
y_pred=gnb.fit(X_train,y_train).predict(X_test)
print(y_pred)
x_new=[[5,5,4,4]]
y_new=gnb.fit(X_train,y_train).predict(x_new)
print("predicted output for [[5,5,4,4]]:",y_new)
print("Naive bayes score :",gnb.score(X_test,y_test))
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/6.py"
[2 1 0 2 0 2 0 1 1 1 1 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 1 1 1 2 0 2 0 0 1 2 2 1 2 1 2 1 1 2 1 1 2 1 2 1 0 2 1 1 1 1 2 0 0 2 1 0 0
 1]
predicted output for [[5,5,4,4]]: [2]
Naive bayes score : 0.9466666666666667

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

Experiment No: 7 REGRESSION TECHNIQUE

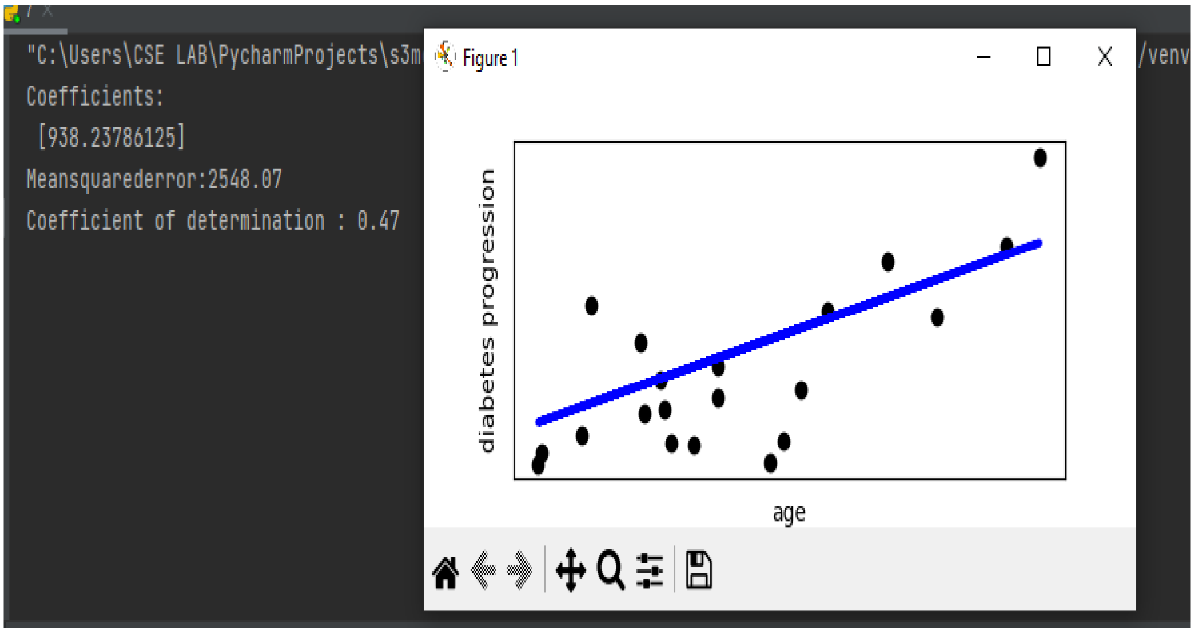
PROGRAM 7.1 LINEAR REGRESSION

AIM : Write a python program to implement linear regression techniques using any standard dataset available in the public domain and evaluate its performance.

PROGRAM

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
df = datasets.load_diabetes()
df['feature_names']
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
diabetes_X.shape
diabetes_y.shape
diabetes_X = diabetes_X[:, np.newaxis, 2]
diabetes_X.shape
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
regr = linear_model.LinearRegression()
regr.fit(diabetes_X_train, diabetes_y_train)
diabetes_y_pred = regr.predict(diabetes_X_test)
print("Coefficients: \n", regr.coef_)
print("Meansquarederror:%.2f"%mean_squared_error (diabetes_y_test, diabetes_y_pred))
print("Coefficient of determination : %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)
plt.xlabel("age")
plt.ylabel("diabetes progression")
plt.xticks(())
plt.yticks(())
plt.show( )
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3m
Coefficients:
 [938.23786125]
Meansquarederror:2548.07
Coefficient of determination : 0.47
```

RESULT

The program has been executed and verified successfully.

PROGRAM 7.2 MULTIPLE REGRESSION

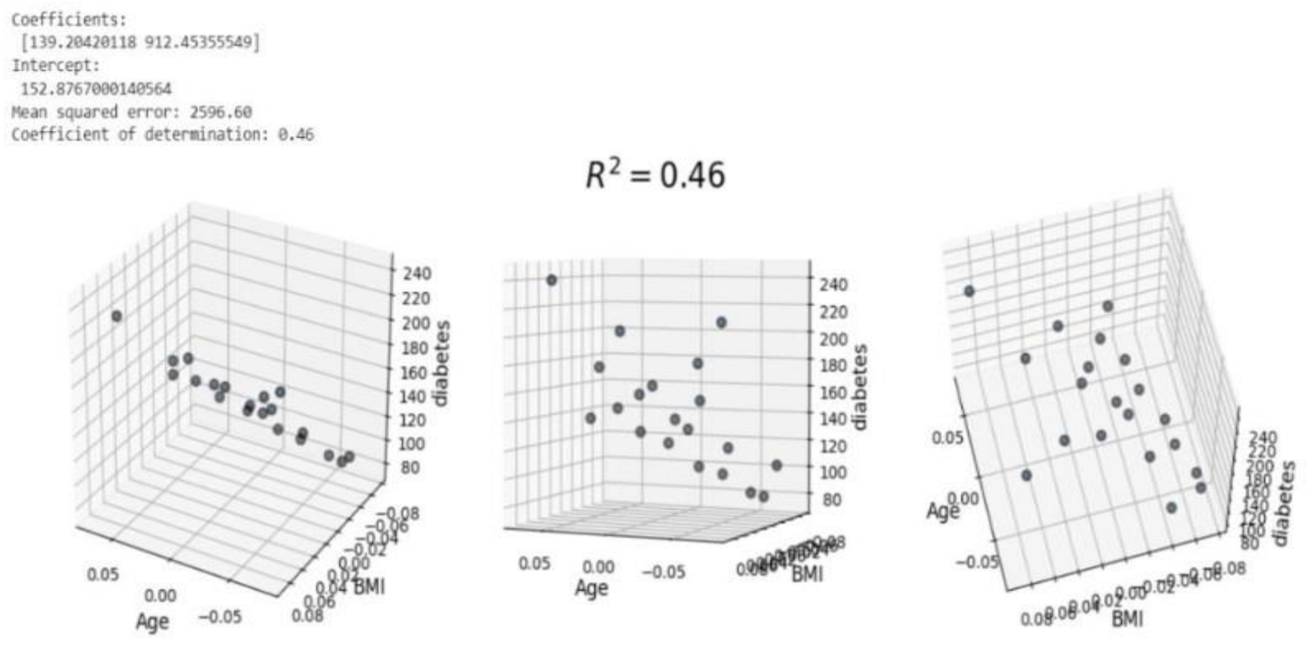
AIM : Write a python program to implement multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.

PROGRAM

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
diabetes_X.shape
diabetes_X = diabetes_X[:,[0,2]]
diabetes_X.shape
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
regr = linear_model.LinearRegression()
regr.fit(diabetes_X_train, diabetes_y_train)
diabetes_y_pred = regr.predict(diabetes_X_test)
print("Coefficients: \n", regr.coef_)
print("Intercept: \n", regr.intercept_)
print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test,
diabetes_y_pred))
print("Coefficient of determination: %.2f" % r2_score(diabetes_y_test, diabetes_y_pred))
x = diabetes_X_test[:, 0]
y = diabetes_X_test[:, 1]
plt.style.use('default')
fig = plt.figure(figsize=(12, 4))
ax1 = fig.add_subplot(120, projection='3d')
ax2 = fig.add_subplot(132, projection='3d')
ax3 = fig.add_subplot(133, projection='3d')
axes = [ax1, ax2, ax3]
```

```
for ax in axes:
ax.plot(x, y, diabetes_y_pred, color='k', zorder=15, linestyle='none', marker='o',alpha=0.5)
ax.scatter(x.flatten(), y.flatten(), diabetes_y_pred, facecolor=(0,0,0,0), s=20,
edgecolor='#70b3f0')
ax.set_xlabel('Age', fontsize=12)
ax.set_ylabel('BMI', fontsize=12)
ax.set_zlabel('diabetes', fontsize=12)
ax.locator_params(nbins=4, axis='x')
ax.locator_params(nbins=5, axis='x')
ax1.view_init(elev=28, azim=120)
ax2.view_init(elev=4, azim=114)
ax3.view_init(elev=60, azim=165)
fig.suptitle('$R^2 = %.2f$' % r2_score(diabetes_y_test, diabetes_y_pred), fontsize=20)
fig.tight_layout()
```

OUTPUT



```
Coefficients:
 [139.20420118 912.45355549]
Intercept:
 152.8767000140564
Mean squared error: 2596.60
Coefficient of determination: 0.46
```

RESULT

The program has been executed and verified successfull

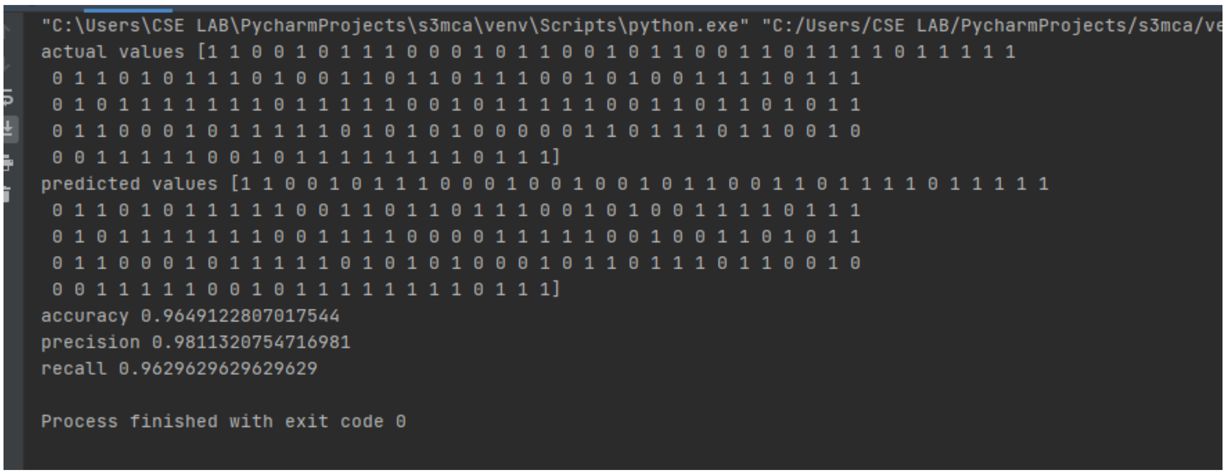
<u>Experiment No: 8</u> <u>SUPPORT VECTOR MACHINES</u>

<u>AIM :</u> Write a python program to implement text classification using Support vector machine.

<u>PROGRAM</u>

```
from sklearn.model_selection import train_test_split
from sklearn import datasets
from sklearn import svm
from sklearn import metrics
cancer=datasets.load_breast_cancer()
x_train,x_test,y_train,y_test=train_test_split(cancer.data,cancer.target,test_size=0.3,random_state=109)
clf=svm.SVC(kernel='linear')
clf.fit (x_train,y_train)
y_pred=clf.predict(x_test)
print("actual values",y_test)
print("predicted values",y_pred)
print("accuracy",metrics.accuracy_score(y_test,y_pred))
print("precision",metrics.precision_score(y_test,y_pred))
print("recall",metrics.recall_score(y_test,y_pred))
```

<u>OUTPUT</u>



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/ve
actual values [1 1 0 0 1 0 1 1 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1
 0 1 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1
 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1
 0 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1 1 0 0 1 0
 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1]
predicted values [1 1 0 0 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 0 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1
 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1 1 1
 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0 1 0 1 1
 0 1 1 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 1 1 0 0 1 0
 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1]
accuracy 0.9649122807017544
precision 0.9811320754716981
recall 0.9629629629629629

Process finished with exit code 0
```

<u>RESULT</u>

The program has been executed and verified successfully.

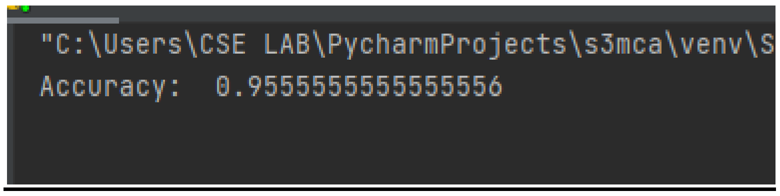
Experiment No: 9 DECISION TREE

AIM : Write a python program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.

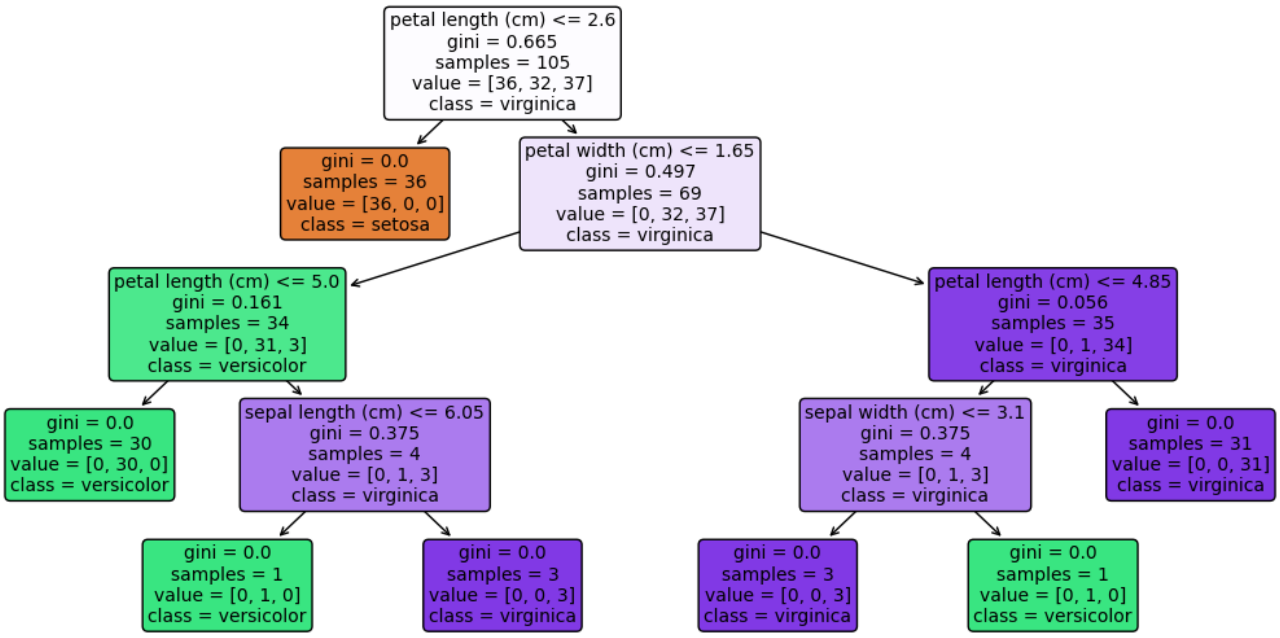
PROGRAM

```
from sklearn.datasets import load_iris

from sklearn import metrics

from sklearn import tree

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

iris=load_iris()

x=iris.data

y=iris.target

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=1)

clf=DecisionTreeClassifier()

clf=clf.fit(x_train,y_train)

y_pred=clf.predict(x_test)

print("Accuracy: ",metrics.accuracy_score(y_test,y_pred))

plt.figure(figsize=(15,15))

tree.plot_tree(clf,fontsize=10,filled=True,rounded=True,class_names=iris.target_names,fa
ture_names=iris.feature_names)

plt.show()
```

OUTPUT



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\S
Accuracy:  0.9555555555555556
```



RESULT

The program has been executed and verified successfully.

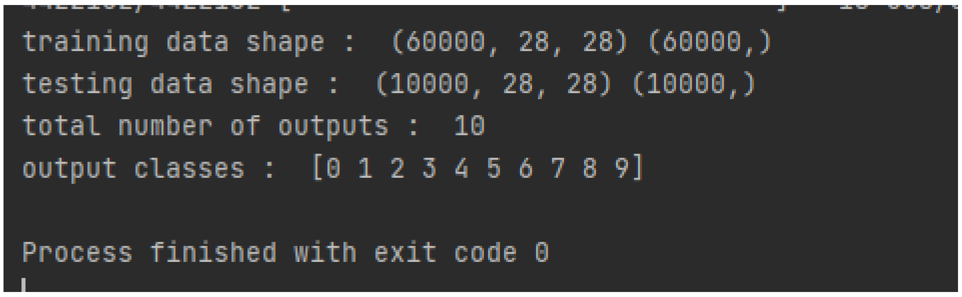
Experiment No: 10 CONVOLUTIONAL NEURAL NETWORK

AIM : Write a python program on convolutional neural network to classify images from any standard datasets in the public domain using Keras framework

PROGRAM

```
import matplotlib
from matplotlib import pyplot as plt
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from keras.datasets import fashion_mnist
(train_X,train_Y), (test_X,test_Y) = fashion_mnist.load_data()
from keras.utils.np_utils import to_categorical
%matplotlib inline
print ('training data shape : ', train_X.shape, train_Y.shape)
print ('testing data shape : ', test_X.shape, test_Y.shape)
('Training data shape : ',(60000, 28, 28), (60000, ))
('Testing data shape : ',(10000, 28, 28), (10000, ))
classes = np.unique(train_Y)
nClasses = len(classes)
print('total number of outputs : ', nClasses)
print('output classes : ', classes)
```

OUTPUT



```
training data shape :  (60000, 28, 28) (60000,)
testing data shape :  (10000, 28, 28) (10000,)
total number of outputs :  10
output classes :  [0 1 2 3 4 5 6 7 8 9]

Process finished with exit code 0
```

RESULT

The program has been executed and verified successfully.

Experiment No: 11 **NATURAL LANGUAGE TOOLKIT**

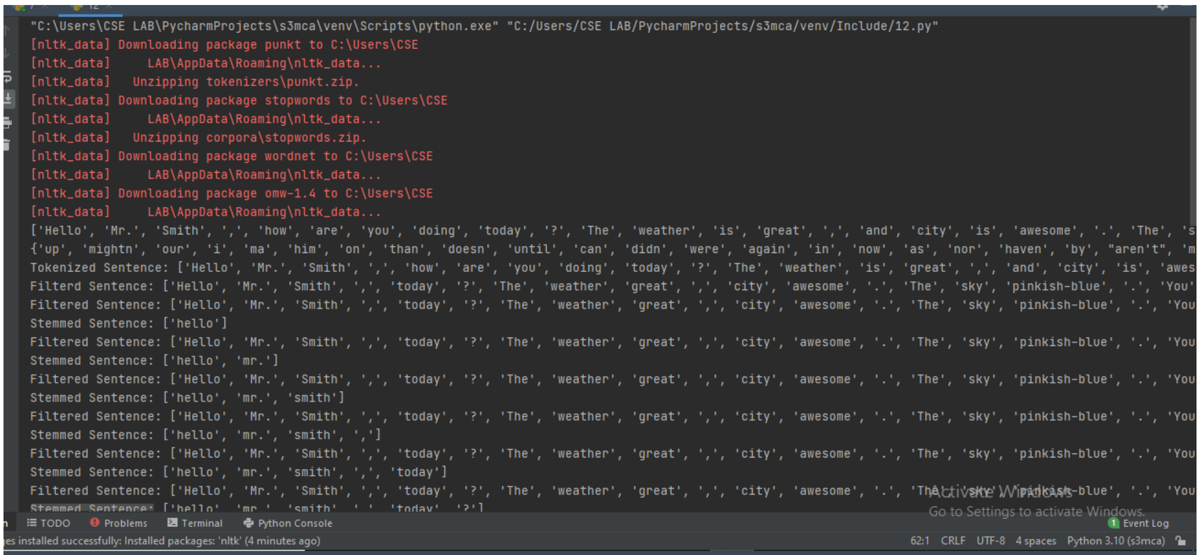
AIM : Write a python program on Natural Language Toolkit.

PROGRAM

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.tokenize import sent_tokenize,word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
text="""Hello Mr. Smith, how are you doing today? The weather is great, and city is
awesome. The sky is pinkish-blue. You shouldn't eat cardboard"""
tokenized_word=word_tokenize(text)
print(tokenized_word)
stop_words=set(stopwords.words("english"))
print(stop_words)
filtered_word=[]
for w in tokenized_word:
if w not in stop_words:
filtered_word.append(w)
print("Tokenized Sentence:",tokenized_word)
print("Filterd Sentence:",filtered_word)
ps = PorterStemmer()
stemmed_words=[]
for w in filtered_word:
stemmed_words.append(ps.stem(w))
print("Filtered Sentence:",filtered_word)
print("Stemmed Sentence:",stemmed_words)
lem = WordNetLemmatizer()
stem = PorterStemmer()
```

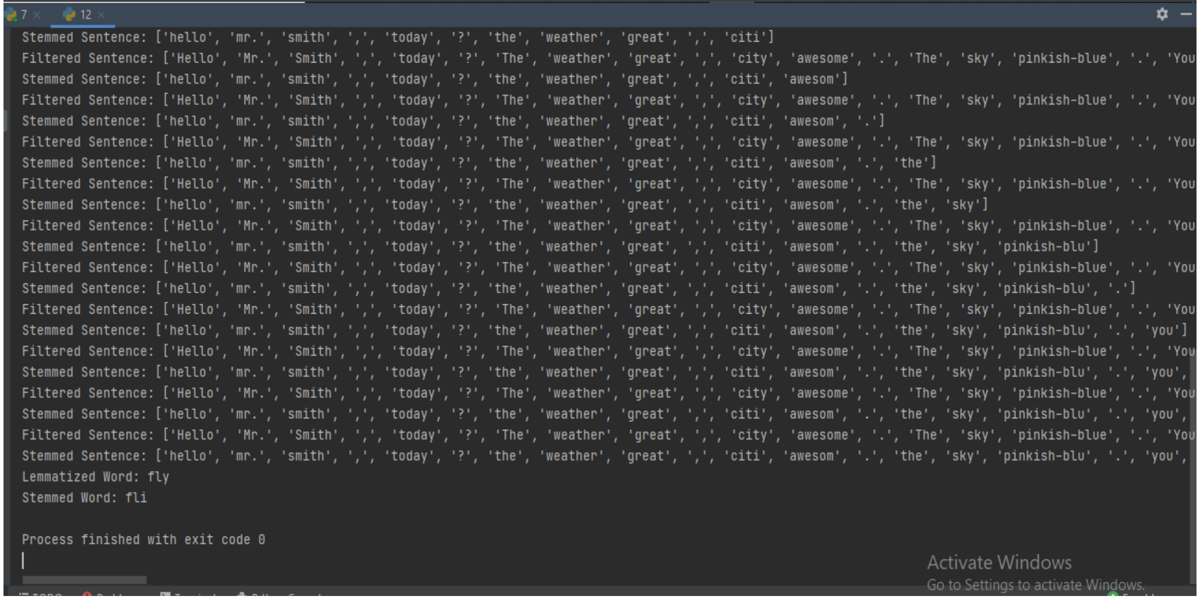
```
word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

<u>OUTPUT</u>



```
"C:\Users\CSE LAB\PycharmProjects\s3mca\venv\Scripts\python.exe" "C:/Users/CSE LAB/PycharmProjects/s3mca/venv/Include/12.py"
[nltk_data] Downloading package punkt to C:\Users\CSE
[nltk_data]     LAB\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
[nltk_data] Downloading package stopwords to C:\Users\CSE
[nltk_data]     LAB\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
[nltk_data] Downloading package wordnet to C:\Users\CSE
[nltk_data]     LAB\AppData\Roaming\nltk_data...
[nltk_data] Downloading package omw-1.4 to C:\Users\CSE
[nltk_data]     LAB\AppData\Roaming\nltk_data...
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'city', 'is', 'awesome', '.', 'The', 's
{'up', 'mightn', 'our', 'i', 'ma', 'him', 'on', 'than', 'doesn', 'until', 'can', 'didn', 'were', 'again', 'in', 'now', 'as', 'nor', 'haven', 'by', "aren't", 'm
Tokenized Sentence: ['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The', 'weather', 'is', 'great', ',', 'and', 'city', 'is', 'awes
Filterd Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You'
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?']
```

TODO
Problems
Terminal
Python Console
Event Log
es installed successfully: Installed packages: 'nltk' (4 minutes ago)
62:1 CRLF UTF-8 4 spaces Python 3.10 (s3mca)



```
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.', 'you']
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.', 'you',
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.', 'you',
Filtered Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?', 'The', 'weather', 'great', ',', 'city', 'awesome', '.', 'The', 'sky', 'pinkish-blue', '.', 'You
Stemmed Sentence: ['hello', 'mr.', 'smith', ',', 'today', '?', 'the', 'weather', 'great', ',', 'citi', 'awesom', '.', 'the', 'sky', 'pinkish-blu', '.', 'you',
Lemmatized Word: fly
Stemmed Word: fli

Process finished with exit code 0
```

<u>RESULT</u>

The program has been executed and verified successfully.