Experiment:3

INTRODUCTION TO PYTHON FOR LINEAR ALGEBRA AND MATRIX OPERATIONS

(Function and NumPy)

1. Write a Python function (naming string reverse) to reverse the given string and then, check whether the string is palindrome or not.

Solution:

```
def string_reverse(s):
    if s[::-1]==s:
        return "yes"
    else:
        return "no"
```

2. Write a function that takes a list of three numbers and returns the largest number that can be obtained by adding any two numbers.

Solution:

```
 def sum_large(x):
    sum1=x[0]+x[1]
    sum2=x[1]+x[2]
    sum3=x[2]+x[0]
    if sum1≥sum2 and sum1≥sum3:
        return sum1
    elif sum2≥sum3:
        return sum2
    else:
        return sum3
```

3. Create the below two matrices:

$$\text{(i)}\quad A = \begin{bmatrix} 1 & -1 & 3 \\ 5 & 7 & 9 \\ -4 & 2 & 8 \end{bmatrix}, \qquad B = \begin{bmatrix} 5 & 7 & 4 \\ -1 & 2 & 5 \\ 0 & 8 & 4 \end{bmatrix}$$

Solution:

```
import numpy as np
A = np.array([[1, -1,3], [5, 7,9],[-4,2,8]])
B = np.array([[5, 7,4], [-1,2, 5],[0,8,4]])
```

4. Find $AB - B^2A$,

   (i) Display the second row and third column of $AB - B^2A$.

   (ii) Find the max and min entry of $AB - B^2A$.

   (iii) Compute the sum of the diagonal entries of $AB - B^2A$.

Solution:

```
Ans=np.matmul(A,B)-np.matmul(np.matmul(B,B),A)
print(Ans)
print(Ans[1,:])#second row
print(Ans[:,2])#third column
print(np.max(Ans))
print(np.min(Ans))
print(np.trace(Ans))
```

5. Show that

$$A^3 - 16A^2 + 70A - 228I = O.$$

Solution:

```
print(np.matmul(np.matmul(A,A),A)-16*np.matmul(A,A)+70*A-228*np.eye(3))
```

6. Write a Python function that takes two matrices as input, then check whether they are compatible with matrix multiplication. If yes, then find their product. {Do it first by using in-build function and then without using in-build function(Means explicitly).}

Solution:

```
def matrix_mul(X,Y):
    m,n1=X.shape;
    n2,p=Y.shape;
    if n1!=n2:
        return "Not possible"
    else:
        C=np.zeros((m,p))
        for i in range(m):
            for j in range(n):
                for k in range(p):
                    C[i,k]=C[i,k]+X[i,j]*Y[j,k]

    return C
```

Solution:

```
  import numpy as np
def multi(A,B):
    if (np.shape(A))[1]==(np.shape(B))[0]:
        print("yes,they are compatible")
        return np.dot(A,B)
    else:
        print("No")
```