Harsh Kasliwal
2003085
C21

Assignment 02

Working with numpy Library

Q1:

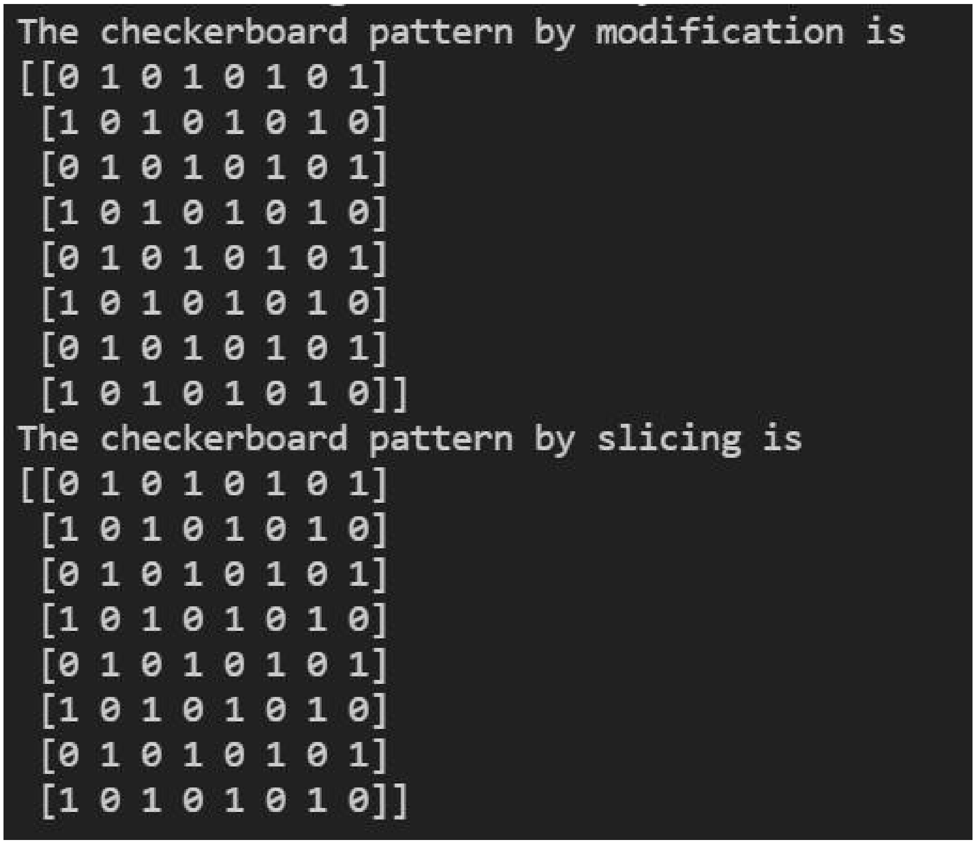
Python program to create a 8x8 matrix and fill it with a checkerboard pattern (slice operator)

Code:

```
import numpy as np


# Creating a all zeros array of size 8x8 arr = np.zeros((8,8),
dtype='int16')


for i in range(8):
    for j in range(8):
        # algorithm for checkerboard pattern arr[i,j] =
        (i+j)%2
print(f'The checkerboard pattern by modification is \n{arr}')
```

```
arr = np.ones((8,8), dtype='int16')

arr[0::2,0:8:2] = 0

arr[1:8:2, 1::2] = 0

# printing the final output

print(f'The checkerboard pattern by slicing is \n{arr}')
```

Output:



```
The checkerboard pattern by modification is
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
The checkerboard pattern by slicing is
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
```

Q2:

Menu driven code for numpy array

- create using array() and arange()
- sum of array
- sort array
- compare two arrays

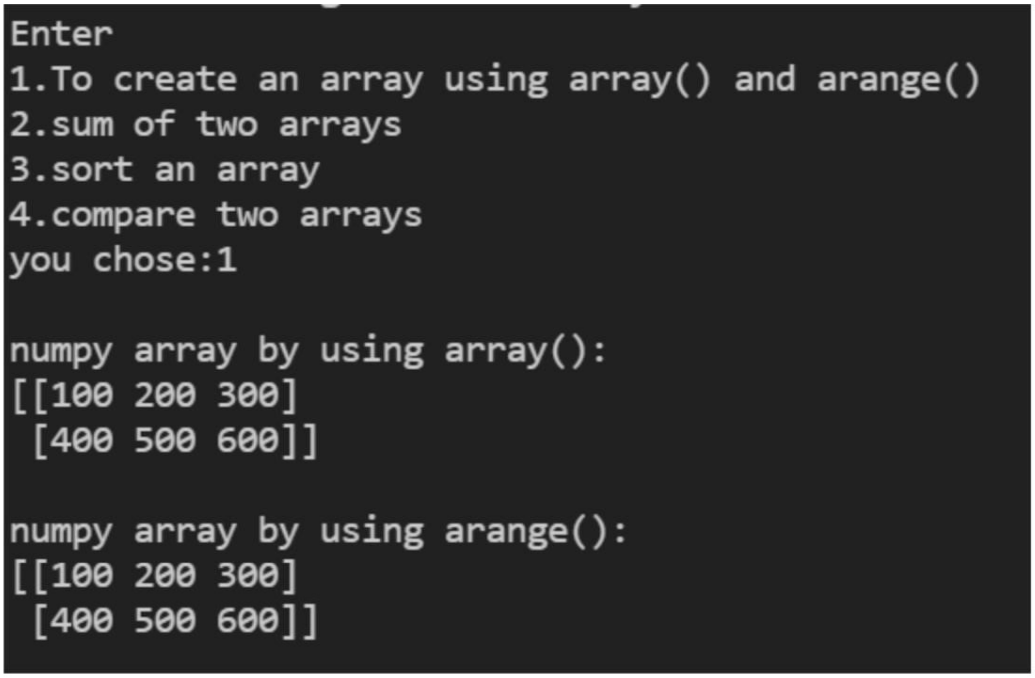
Code:

```
import numpy as np

# asking what he wants us to do

inp = int(input("Enter\n1.To create an array using array() and arange()\n2.sum of two arrays\n3.sort an
array\n4.compare two arrays\nyou chose:"))
# if user chooses 1 if inp
== 1:

    # creating a list to put it inside array() method list1 =

    [[100,200,300],[400,500,600]] first_arr = np.array(list1,dtype='int64')

    print(f"\nnumpy array by using array():\n{first_arr}")


    # making the same array by using arange() with start=100,stop=700(exclusive)
step=100

    # reshaping the 1d array to 2x3 array

    second_arr = np.arange(start=100,stop=700,step=100).reshape(2,3)

    print(f"\nnumpy array by using arange():\n{second_arr}")


# if user chooses 2 elif

inp == 2:

    # creating a random array of size 3X3 of range 10 first_arr =

    np.random.randint(10,size =(3,3)) print(f"The first array is

    \n{first_arr}")
```

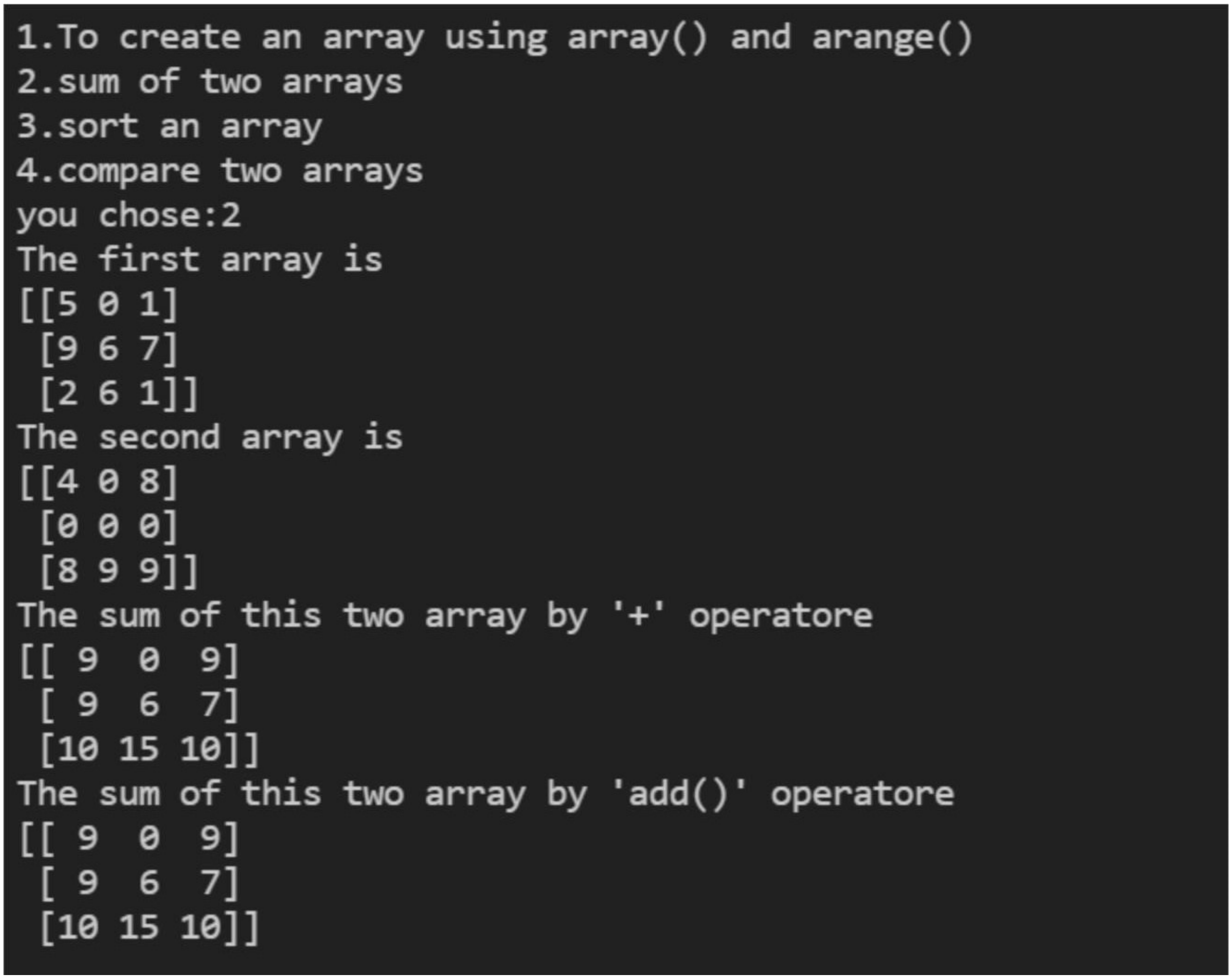
```
        # creating a random array of size 3X3 of range 10 second_arr =

        np.random.randint(10,size =(3,3)) print(f"The second array is

        \n{second_arr}")

        # adding the two arrays by + and add()

        print(f"The sum of this two array by '+' operatore \n{first_arr + second_arr}")


        print(f"The sum of this two array by 'add()' operatore
\n{np.add(first_arr,second_arr)}")


# if user chooses 3 elif

inp == 3:

        # creating a random array of size 3X3 of range 6 first_arr =

        np.random.randint(6,size =(3,3)) print(f"The array is \n{first_arr}")


        # sorting the array by using np.sort()

        print(f"The array after sorting is :\n{np.sort(first_arr)}")
elif inp == 4:

        print("The first set of arrays are\n")

        # creating a random array of size 3X3 of range 6 first_arr =

        np.random.randint(6,size =(3,3)) print(f"The first array is

        \n{first_arr}")

        # copying the first array in the second array second_arr = np.copy(first_arr)

        print(f"The second array is \n{second_arr}")


           print(f"Are this two arrays
equal?\nans:{np.array_equiv(first_arr,second_arr)}\n")

        print("The second set of arrays are\n")

        # creating a random array of size 3X3 of range 6 first_arr =

        np.random.randint(6,size =(3,3)) print(f"The first array is

        \n{first_arr}")

        # copying the first array in the second array second_arr = np.copy(first_arr)
```

```
    # changing any one value

    second_arr[0,2] = 5

    print(f"The second array is \n{second_arr}")

      print(f"Are this two arrays
equal?\nans:{np.array_equiv(first_arr,second_arr)}\n")


else:

    print("Select a valid option:")
```

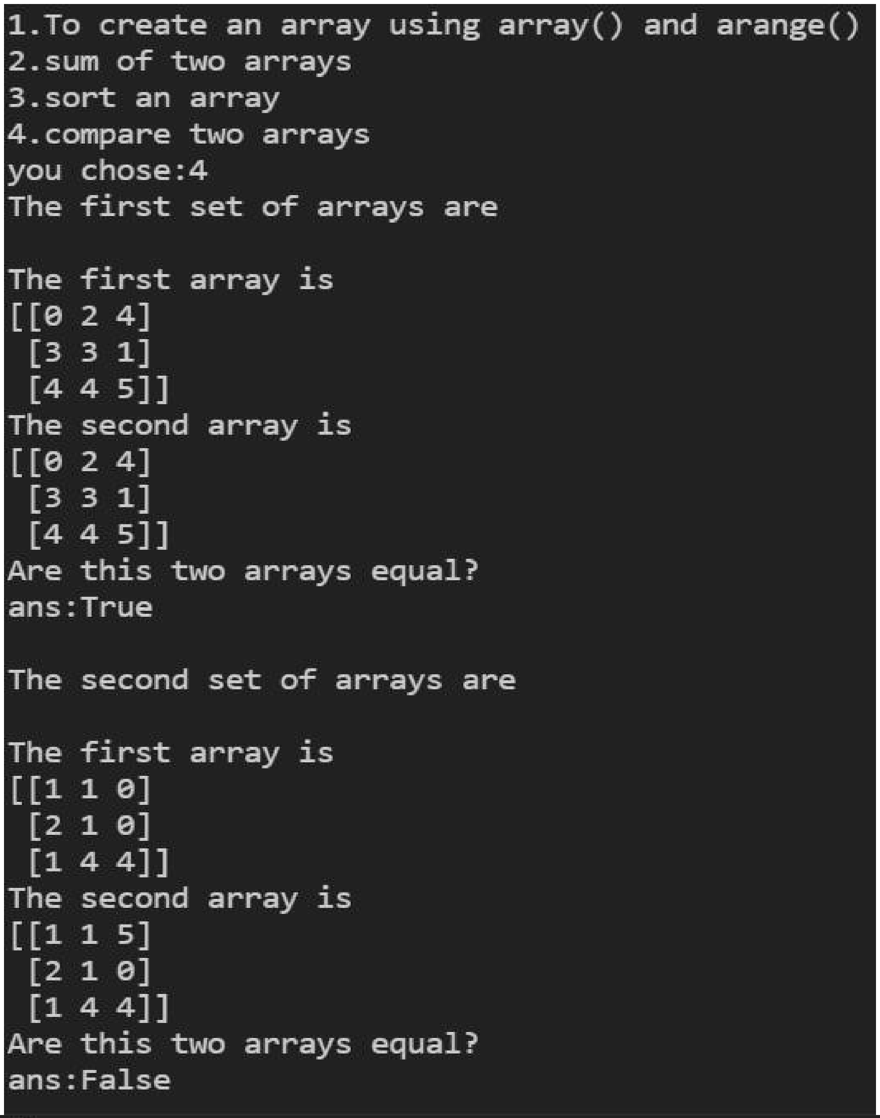
OUTPUT:



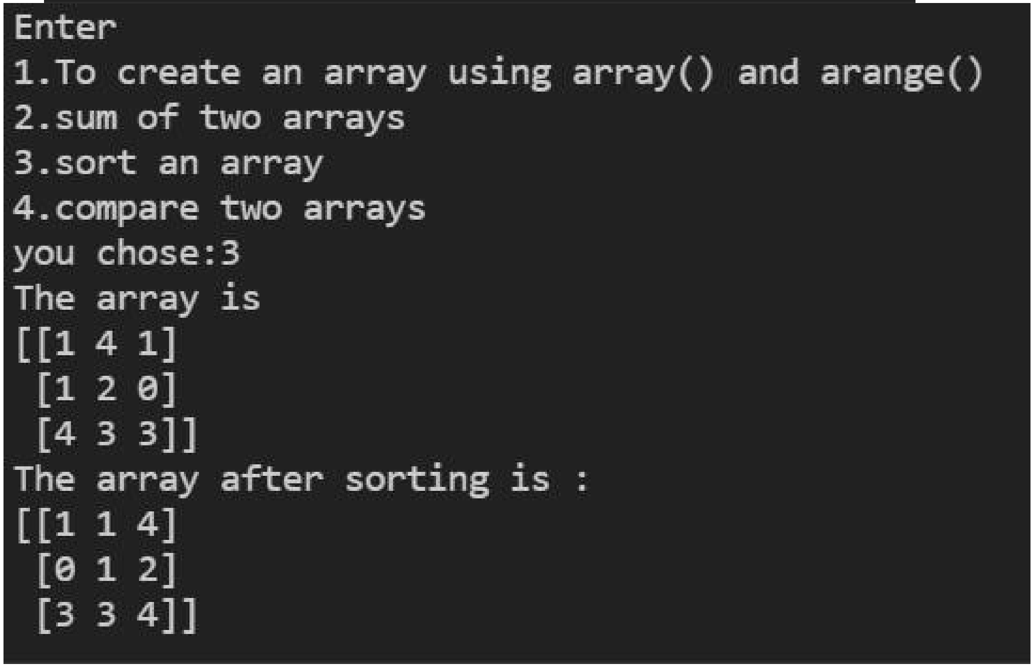
```
Enter
1.To create an array using array() and arange()
2.sum of two arrays
3.sort an array
4.compare two arrays
you chose:1

numpy array by using array():
[[100 200 300]
 [400 500 600]]

numpy array by using arange():
[[100 200 300]
 [400 500 600]]
```



```
1.To create an array using array() and arange()
2.sum of two arrays
3.sort an array
4.compare two arrays
you chose:2
The first array is
[[5 0 1]
 [9 6 7]
 [2 6 1]]
The second array is
[[4 0 8]
 [0 0 0]
 [8 9 9]]
The sum of this two array by '+' operatore
[[ 9  0  9]
 [ 9  6  7]
 [10 15 10]]
The sum of this two array by 'add()' operatore
[[ 9  0  9]
 [ 9  6  7]
 [10 15 10]]
```



```
1.To create an array using array() and arange()
2.sum of two arrays
3.sort an array
4.compare two arrays
you chose:4
The first set of arrays are

The first array is
[[0 2 4]
 [3 3 1]
 [4 4 5]]
The second array is
[[0 2 4]
 [3 3 1]
 [4 4 5]]
Are this two arrays equal?
ans:True

The second set of arrays are

The first array is
[[1 1 0]
 [2 1 0]
 [1 4 4]]
The second array is
[[1 1 5]
 [2 1 0]
 [1 4 4]]
Are this two arrays equal?
ans:False
```



```
Enter
1.To create an array using array() and arange()
2.sum of two arrays
3.sort an array
4.compare two arrays
you chose:3
The array is
[[1 4 1]
 [1 2 0]
 [4 3 3]]
The array after sorting is :
[[1 1 4]
 [0 1 2]
 [3 3 4]]
```

Q3:

Python program

- To read a two matrices from user
- Perform matrix multiplication
- Display diagonal Elements
- Check whether its a square matrix

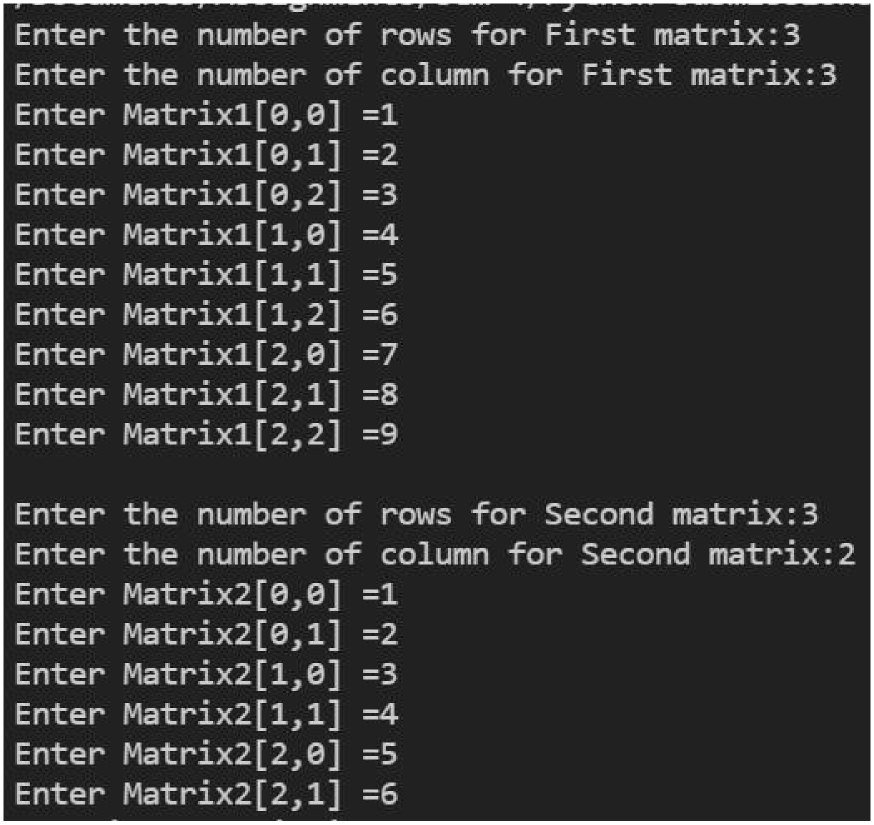
Code:

```
import numpy as np


# asking the shape of matrix (row x column)

row = int(input(("Enter the number of rows for First matrix:")))

column = int(input(("Enter the number of column for First matrix:")))

# making all zero matrix of shape row x column matrix1 =

np.zeros((row,column),dtype='int32') for i in range(row):


        for j in range(column):

                # modifing the values as per user input.
 matrix1[i,j] = int(input(f"Enter Matrix1[{i},{j}] ="))


print()

# asking the shape of matrix (row x column)

row = int(input(("Enter the number of rows for Second matrix:")))

column = int(input(("Enter the number of column for Second matrix:")))
```

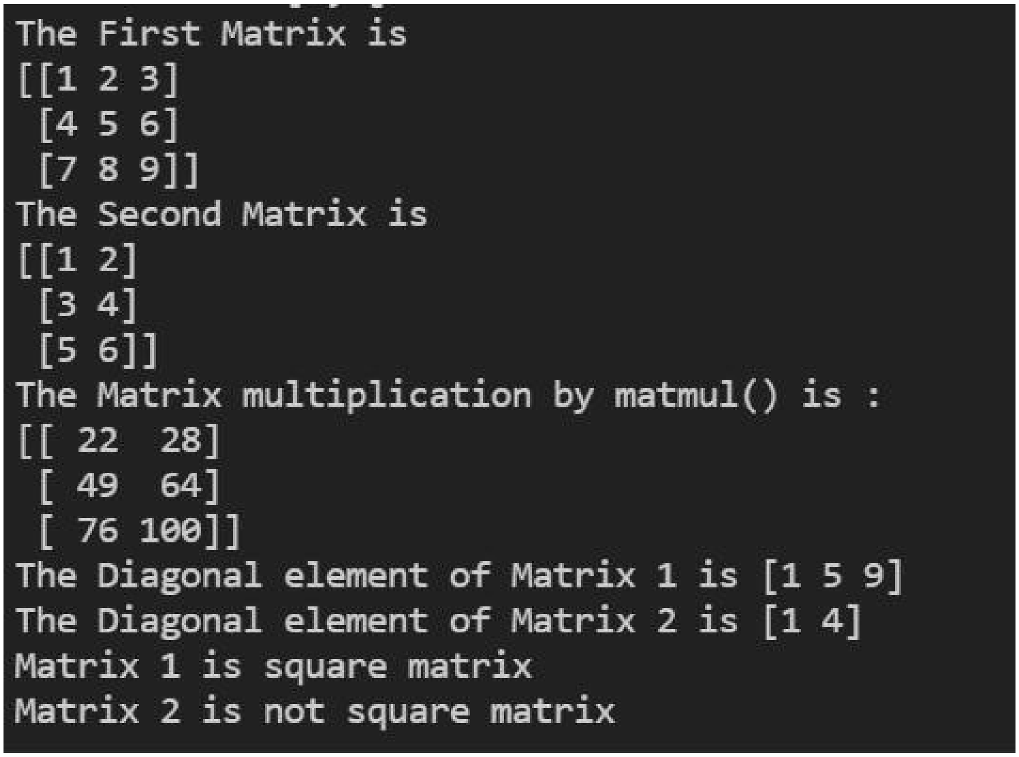
```
# making all zero matrix of shape row x column matrix2 =

np.zeros((row,column),dtype='int32') for i in range(row):


    for j in range(column):

        # modifing the values as per user input.
 matrix2[i,j] = int(input(f"Enter Matrix2[{i},{j}] ="))


print(f"The First Matrix is \n{matrix1}")

print(f"The Second Matrix is \n{matrix2}")


# matrix multiplication by np.matmul()

print(f"The Matrix multiplication by matmul() is
:\n{np.matmul(matrix1,matrix2)}")
print(f"The Diagonal element of Matrix 1 is {np.diagonal(matrix1)}\nThe Diagonal
element of Matrix 2 is {np.diagonal(matrix2)}")



# taking the shape of matrix 1 and determining if it is square or not
rows,cols =

matrix1.shape

if rows == cols:

    print("Matrix 1 is square matrix") else:


    print("Matrix 1 is not square matrix")



# taking the shape of matrix 1 and determining if it is square or not
rows,cols =

matrix2.shape

if rows == cols:

    print("Matrix 2 is square matrix") else:
```

```
print("Matrix 2 is not square matrix")
```

Output:



```
Enter the number of rows for First matrix:3
Enter the number of column for First matrix:3
Enter Matrix1[0,0] =1
Enter Matrix1[0,1] =2
Enter Matrix1[0,2] =3
Enter Matrix1[1,0] =4
Enter Matrix1[1,1] =5
Enter Matrix1[1,2] =6
Enter Matrix1[2,0] =7
Enter Matrix1[2,1] =8
Enter Matrix1[2,2] =9

Enter the number of rows for Second matrix:3
Enter the number of column for Second matrix:2
Enter Matrix2[0,0] =1
Enter Matrix2[0,1] =2
Enter Matrix2[1,0] =3
Enter Matrix2[1,1] =4
Enter Matrix2[2,0] =5
Enter Matrix2[2,1] =6
```



```
The First Matrix is
[[1 2 3]
 [4 5 6]
 [7 8 9]]
The Second Matrix is
[[1 2]
 [3 4]
 [5 6]]
The Matrix multiplication by matmul() is :
[[ 22  28]
 [ 49  64]
 [ 76 100]]
The Diagonal element of Matrix 1 is [1 5 9]
The Diagonal element of Matrix 2 is [1 4]
Matrix 1 is square matrix
Matrix 2 is not square matrix
```

Q4:

Delete the second column from a given array and insert the following new column in its place.

sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])

newColumn = numpy.array([[1,1,1]])

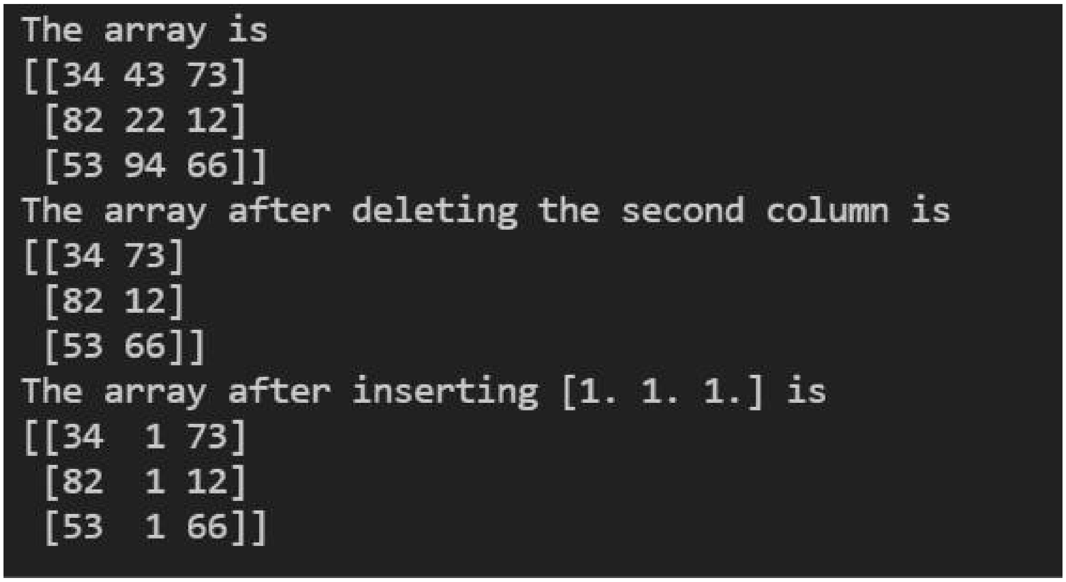
Array:[[34 43 73][82 22 12][53 94 66]]

Expected output:[[34 173][82 112][53 166]]

Code:

```
import numpy as np


arr = np.array([[34,43,73],[82,22,12],[53,94,66]])

column = np.ones(3)


print(f"The array is \n{arr}")
# deleting the second column arr =
np.delete(arr,1,1)

print(f"The array after deleting the second column is \n{arr}")



# inserting a column at second place

arr = np.insert(arr,1,column,1)

print(f"The array after inserting {column} is \n{arr}")
```

Output:



```
The array is
[[34 43 73]
 [82 22 12]
 [53 94 66]]
The array after deleting the second column is
[[34 73]
 [82 12]
 [53 66]]
The array after inserting [1. 1. 1.] is
[[34  1 73]
 [82  1 12]
 [53  1 66]]
```