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')

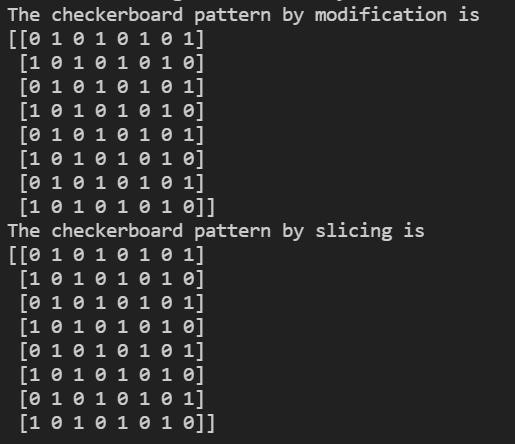
arr[0::2,0:8:2] = 0

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

# printing the final output

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

**Output:**



**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

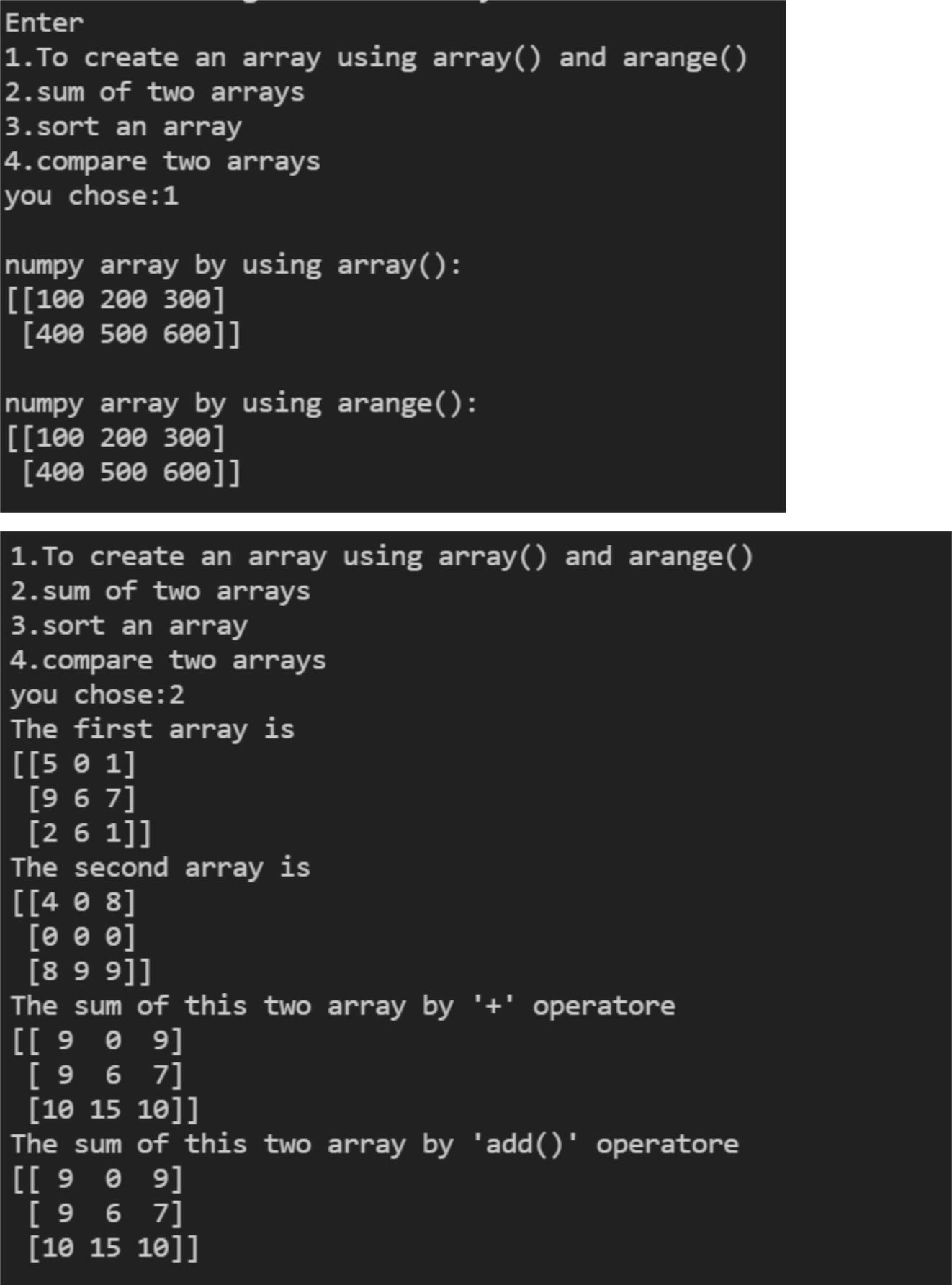
print(f"The second array is \n{second\_arr}")

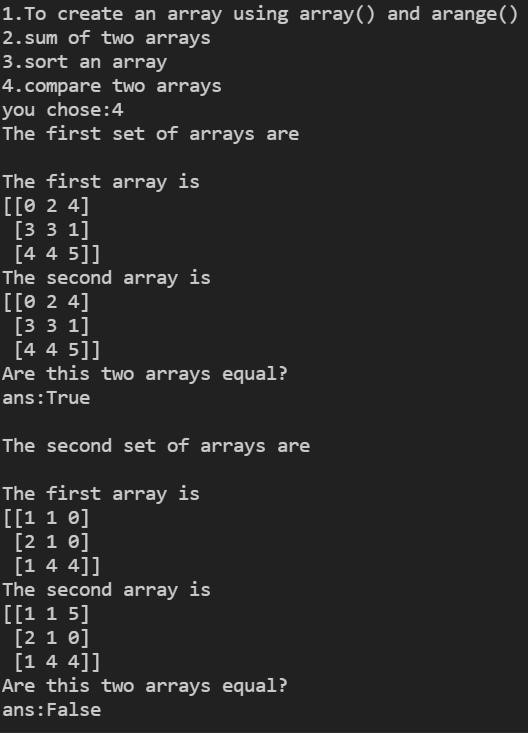
print(f"Are this two arrays equal?\nans:{np.array\_equiv(first\_arr,second\_arr)}\n")

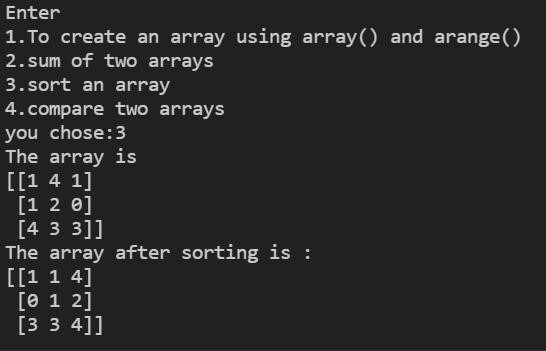
else:

print("Select a valid option:")

**OUTPUT:**







**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 =

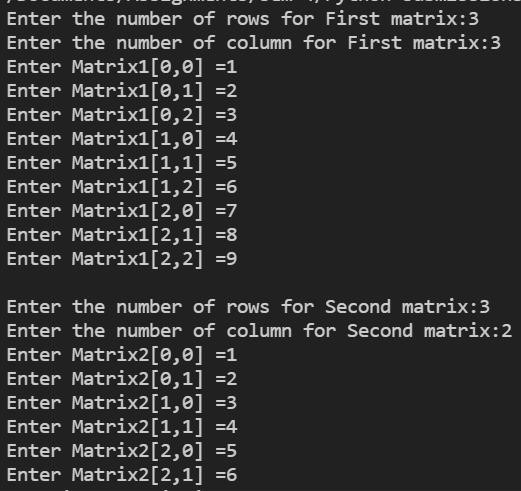
matrix2.shape

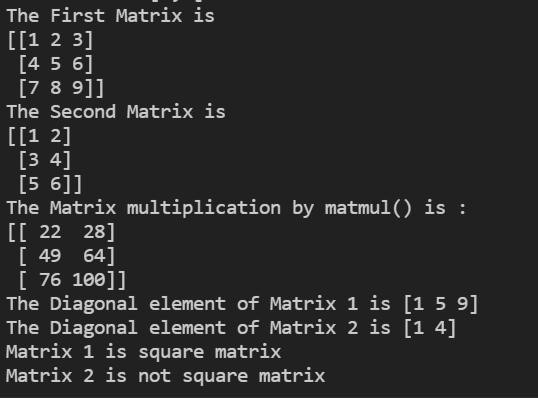
if rows == cols:

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

print("Matrix 2 is not square matrix")

**Output:**





**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:**

