1. **To generate the random numbers by using different generator parameters and create a file as database to be used for realization of matrix operations**

|  |
| --- |
|  |
|  | import numpy as np  import os |
|  | from numpy import random |
|  |  |
|  | mat = random.randint(100,size=(10,10)) |
|  |  |
|  | np.savetxt("matrix.txt",mat) |

1. **Perform the matrix basic operations (Addition, Subtraction Multiply, Matrix Inversion, (a matrix and a vector)), for different dimensions of the matrix.**

**COMBINED(Q2 & Q3)**

1. **Multiply 2 matrices A[2000,2000] and B[2000,6000]. Calculate the computational time.**

import numpy as np

import random

import time

t=time.time()

x=[[random.randint(1,100) for i in range(2000)] for j in range(2000)]

y=[[random.randint(1,100) for i in range(2000)] for j in range(2000)]

print(np.add(x, y))

print(np.subtract(x, y))

print(np.divide(x, y))

print(np.linalg.inv(x))

print(np.linalg.inv(y))

print(np.dot(x,y))

print(time.time()-t)

1. **Perform parallel program for the matrix operations (Addition, Multiply, (matrix and vector)). Calculate the computational time.**

|  |
| --- |
| from multiprocessing import Process |
|  | import time |
|  | from numpy import random |
|  | import numpy as np |
|  |  |
|  | # \_\_init\_\_ method you can initialize resource and by implementing Process.run() method you can write the code for the subprocess. |
|  |  |
|  | a=random.randint(100,size=(10,10)) |
|  | b=random.randint(100,size=(10,10)) |
|  |  |
|  | def subtract(): |
|  | print('subtract\n') |
|  | print(np.subtract(a,b)) |
|  | def multiply(): |
|  | print('multiply\n') |
|  | print(np.multiply(a,b)) |
|  |  |
|  | def add(): |
|  | print('add\n') |
|  | print(np.add(a,b)) |
|  |  |
|  | def divide(): |
|  | print('divide\n') |
|  | print(np.divide(a,b)) |
|  |  |
|  |  |
|  |  |
|  |  |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  |  |
|  | p = Process(target=add) |
|  | p.start() |
|  | p.join() |
|  | p2 = Process(target=subtract) |
|  | p2.start() |
|  | p2.join() |
|  | p3 = Process(target=multiply) |
|  | p3.start() |
|  | p3.join() |
|  | p4 = Process(target=divide()) |
|  | p4.start() |
|  | p4.join() |
|  |  |