JOY OF COMPUTING USING PYTHON

Experiment-1

Date:03/09/2018

**Objective**:Write a program to find the sum and difference of two matrices.

**Code**:

import random

def make\_matrix(n):

return [[random.randint(1,9) for i in range(n)] for i in range(n)]

def print\_matrix(matrix, n):

for i in range(n):

for j in range(n):

print(matrix[i][j],end=" ")

print()

print()

def operate\_matrix(matrix1, matrix2, n, func):

resultant\_matrix = []

for i in range(n):

temp\_matrix = []

for j in range(n):

temp\_matrix.append(func(matrix1[i][j],matrix2[i][j]))

resultant\_matrix.append(temp\_matrix)

return resultant\_matrix

add\_elements = lambda a, b: a + b

sub\_elements = lambda a, b: a - b

n = 3

matrix1=make\_matrix(3)

matrix2=make\_matrix(3)

add\_matrix = operate\_matrix(matrix1,matrix2,n, add\_elements)

sub\_matrix = operate\_matrix(matrix1,matrix2,n, sub\_elements)

print("Matrix 1 : ")

print\_matrix(matrix1, n)

print("Matrix 2 : ")

print\_matrix(matrix2, n)

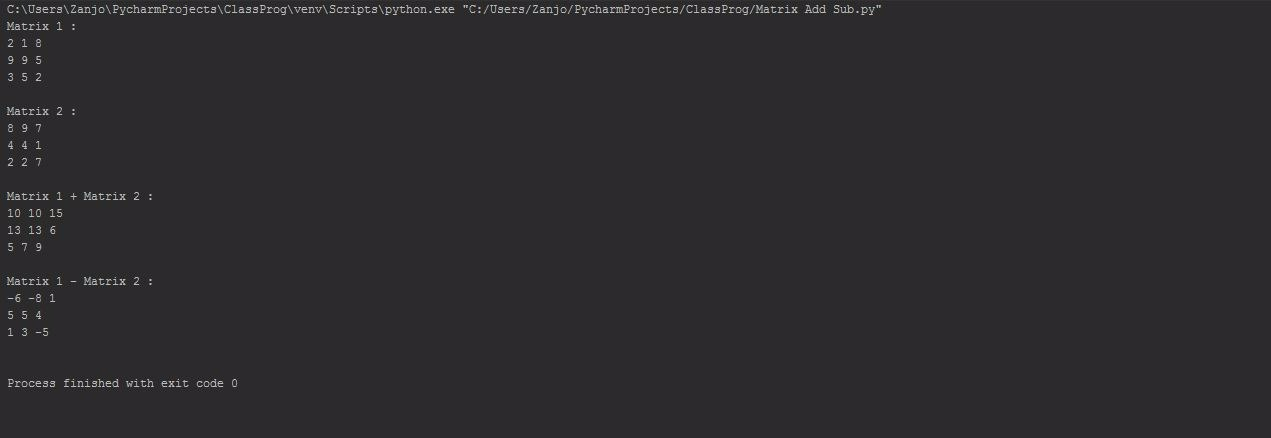
print("Matrix 1 + Matrix 2 : ")

print\_matrix(add\_matrix, n)

print("Matrix 1 - Matrix 2 : ")

print\_matrix(sub\_matrix, n)

Output :



Experiment-2

**Objective**:Write a program to find the multiplication of N number of matrices (same size and different size).

**Code**:

import random

def generate\_matrix(r1,c1):

matrix1 = []

for i in range(r1):

temp = []

for j in range(c1):

temp.append(random.randint(1, 9))

matrix1.append(temp)

return matrix1

def print\_matrix(matrix, r1,c1):

for i in range(r1):

for j in range(c1):

print(matrix[i][j],end=" ")

print()

print()

def multiply\_matrices(matrix1, r1, c1,matrix2,c2):

result = []

for i in range(r1):

temp =[]

for j in range(c2):

sum = 0

for k in range(c1):

sum+= matrix1[i][k] \* matrix2[k][j]

temp.append(sum)

result.append(temp)

return result

# Multiplication of 2 matrices

print("Performing multiplication of 2 matrices.")

matrix1, matrix2 = generate\_matrix(2,2),generate\_matrix(2,2)

print("A :- ")

print\_matrix(matrix1,2,2)

print("B :- ")

print\_matrix(matrix2,2,2)

resultant\_matrix = multiply\_matrices(matrix1,2,2,matrix2,2)

print("A \* B :- ")

print\_matrix(resultant\_matrix,2,2)

# Multiplication 3 different matrices of different sizes

print("Performing multiplication of 3 matrices of different sizes. ")

matrix3, matrix4, matrix5 = generate\_matrix(2,3), generate\_matrix(3,2), generate\_matrix(2,3)

print("C :-")

print\_matrix(matrix3,2,3)

print("D :-")

print\_matrix(matrix4,3,2)

print("E :-")

print\_matrix(matrix5,2,3)

resultant\_matrix2= multiply\_matrices(matrix3,2,3,matrix4,2)

print("Performing (C \* D) \* E :-")

print("(C \* D) :-")

print\_matrix(resultant\_matrix2,2,2)

resultant\_matrix2= multiply\_matrices(resultant\_matrix2,2,2,matrix5,3)

print("(C \* D) \* E :-")

print\_matrix(resultant\_matrix2,2,3)

print("Performing C \* (D \* E) :- ")

resultant\_matrix2 = multiply\_matrices(matrix4,3,2,matrix5,3)

print("(D \* E) :-")

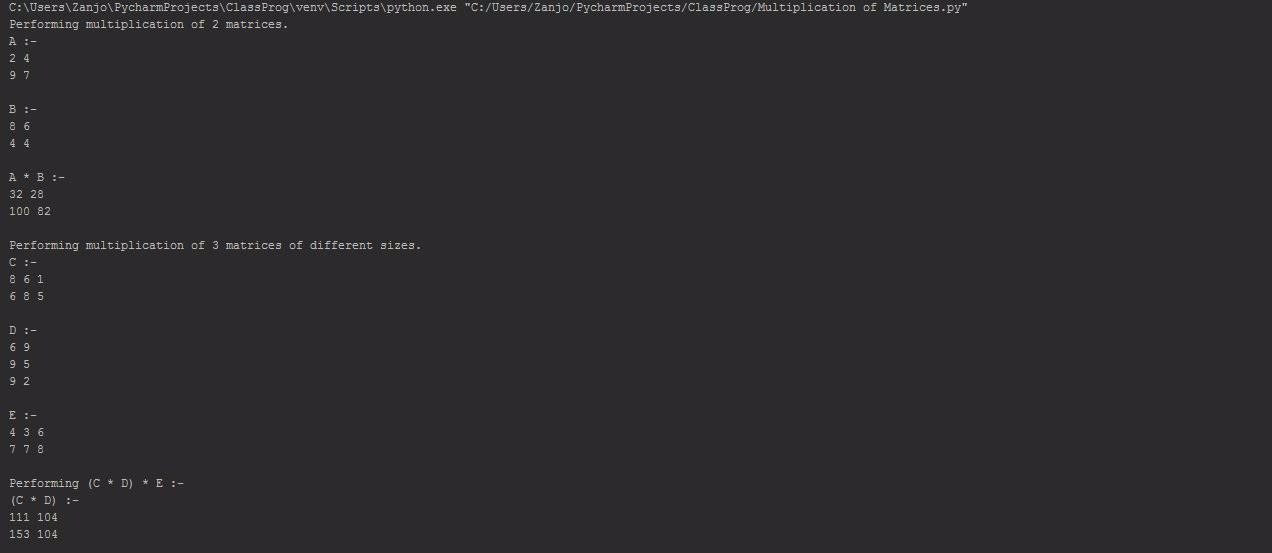
print\_matrix(resultant\_matrix2,3,3)

resultant\_matrix2 = multiply\_matrices(matrix3,2,3,resultant\_matrix2,3)

print("C \* (D \* E) :-")

print\_matrix(resultant\_matrix2,2,3)

OUTPUT:





Experiment-3

**Objective**:Write a program to find the greatest common divisor and lowest common multiple of given two numbers.

**Code**:

def find\_greatest\_common\_divisor(n1, n2):

while n1!=n2:

if n1 > n2:

n1 -=n2

else:

n2 -= n1

return n1

def find\_lowest\_common\_multiple(n1, n2, gcd):

return (n1\*n2)//gcd

a = 60

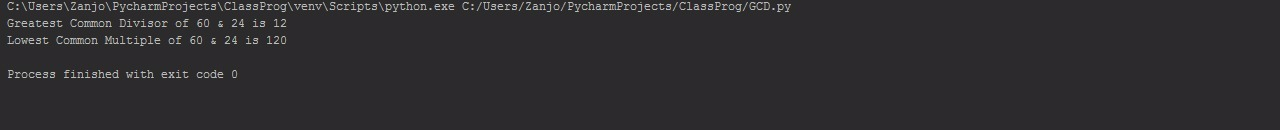
b = 24

gcd = find\_greatest\_common\_divisor(a,b)

print("Greatest Common Divisor of",a,"&",b,"is",gcd)

print("Lowest Common Multiple of",a,"&",b,"is",find\_lowest\_common\_multiple(a, b, gcd))

OUTPUT:



Experiment-4

**Objective**:Write a program to use all trigonometric functions.

**Code**:

import math

pi = math.pi

print("Using the sin function:",math.sin(pi))

print("Using the cos function:",math.cos(pi))

print("Using the tan function:",math.tan(pi))

print("Using the sec function:",1/math.sin(pi))

print("Using the cosec function:",1/math.cos(pi))

print("Using the cot function:",1/math.tan(pi))

OUTPUT:

