**MUHAMMAD SHERAZ**

**BSDSF21A002**

**LAB 10**

**PROGRAMMING FUNDAMENTALS**

**TASK 1**

class matrix:

pass

def create(rows,cols):

m = matrix()

m.rows = rows

m.cols = cols

m.data = [[ 0 for i in range(cols)] for i in range(rows)]

return m

def inp(m):

i = 0

while i < m.rows:

j = 0

while j < m.cols:

m.data[i][j] = int(input())

j += 1

i += 1

return m

def printmatrix(m):

i = 0

while i < m.rows:

j = 0

while j < m.cols:

print(m.data[i][j],end=" ")

j += 1

print()

i += 1

def transpose(m):

t = create(m.rows,m.cols)

i = 0

while i < m.rows:

j = 0

while j < m.cols:

t.data[i][j] = m.data[j][i]

j += 1

i += 1

return t

def add(m1,m2):

if m1.rows != m2.rows or m1.cols != m2.cols:

raise Exception ("for addition rows and colums both should be equal")

t = create(m1.rows,m1.cols)

i = 0

while i < m1.rows:

j = 0

while j < m1.cols:

t.data[i][j] = m1.data[i][j] + m2.data[i][j]

j += 1

i += 1

return t

def issymetric(m):

i = 0

while i < m.rows:

j = 0

while j < m.cols:

if m.data[i][j] != m.data[j][i]:

return False

else:

j += 1

i += 1

return True

def main():

matrix1 = create(2,2)

matrix2 = create(2,2)

print("null matrix=")

printmatrix(matrix1)

print("type elements of matrix 1")

inpm1= inp(matrix1)

print("type elements of matrix 2")

inpm2 = inp(matrix2)

print("matrix 1=")

printmatrix(inpm1)

print("matrix 2=")

printmatrix(inpm2)

addition = add(inpm1,inpm2)

print("addition of matrixes=")

printmatrix(addition)

transpose\_of\_m1 = transpose(inpm1)

transpose\_of\_m2 = transpose(inpm2)

print("transpose of matrix 1 =")

printmatrix(transpose\_of\_m1)

print("transpose of matrix 2 =")

printmatrix(transpose\_of\_m2)

print("matrix 1 symmetric =",issymetric(inpm1))

print("matrix 2 symmetric =",issymetric(inpm2))

main()

**TASK2**

**class matrix:**

**pass**

**def create(rows,cols):**

**m = matrix()**

**m.rows = rows**

**m.cols = cols**

**m.data = [[ 0 for i in range(cols)] for i in range(rows)]**

**return m**

**def identitymatrix(m):**

**if m.rows != m.cols:**

**raise Exception ("for identity matrix m.rows == m.cols")**

**j = 0**

**i = 0**

**while j < m.cols:**

**m.data[i][j] = 1**

**j += 1**

**i += 1**

**return m**

**def inp(m):**

**i = 0**

**while i < m.rows:**

**j = 0**

**while j < m.cols:**

**m.data[i][j] = int(input())**

**j += 1**

**i += 1**

**return m**

**def printmatrix(m):**

**i = 0**

**while i < m.rows:**

**j = 0**

**while j < m.cols:**

**print(m.data[i][j],end=" ")**

**j += 1**

**print()**

**i += 1**

**def subt(m1,m2):**

**if m1.rows != m2.rows or m1.cols != m2.cols:**

**raise Exception ("for subtractio rows and colums both should be equal ")**

**t = create(m1.rows,m1.cols)**

**i = 0**

**while i < m1.rows:**

**j = 0**

**while j < m1.cols:**

**t.data[i][j] = m1.data[i][j] - m2.data[i][j]**

**j += 1**

**i += 1**

**return t**

**def mul(m1,m2):**

**if m1.cols != m2.rows:**

**raise Exception ("for multiolicatio m1.cols = m2.rows")**

**t = create(m1.cols,m2.rows)**

**i = 0**

**while i < m1.cols:**

**j = 0**

**while j < m2.rows:**

**m = 0**

**while m < m2.cols:**

**t.data[i][j] = m1.data[i][j] \* m2.data[j][m]**

**m += 1**

**j += 1**

**i += 1**

**return t**

**def issymetric(m):**

**i = 0**

**while i < m.rows:**

**j = 0**

**while j < m.cols:**

**if m.data[i][j] != m.data[j][i]:**

**return False**

**else:**

**j += 1**

**i += 1**

**return True**

**def issquare(m):**

**if m.rows == m.cols:**

**return True**

**else:**

**return False**

**def main():**

**matrix1 = create(2,2)**

**matrix2 = create(2,2)**

**print("type elements of matrix 1")**

**inpm1= inp(matrix1)**

**print("type elements of matrix 2")**

**inpm2 = inp(matrix2)**

**matrix = create(2,2)**

**identity= identitymatrix(matrix)**

**print("identity matrix=")**

**printmatrix(identity)**

**print("matrix 1=")**

**printmatrix(inpm1)**

**print("matrix 2=")**

**printmatrix(inpm2)**

**subtraction = subt(inpm1,inpm2)**

**print("subtraction of matrixes=")**

**printmatrix(subtraction)**

**print("matrix1 is square=",issquare(inpm1))**

**print("matrix2 is square=",issquare(inpm2))**

**multiplication = mul(inpm1,inpm2)**

**print("multiplication of matrixes=")**

**printmatrix(multiplication)**

**main()**

**TASK3**

from random import randint

def create(x,y,z):

a = [[[0 for i in range(z)] for i in range(y)] for i in range(x)]

return a

def inputmatrix(m):

i = 0

while i < 4:

j = 0

while j < 9:

k = 0

while k < 7:

m[i][j][k] = randint(11,99)

k += 1

j += 1

i += 1

return m

def input2Dmatrix(m):

t = [[0 for c in range(7)] for r in range(9)]

i = 0

while i < 4:

j = 0

while j < 9:

k = 0

while k < 7:

t[j][k] = m[i][j][k]

k += 1

j += 1

i += 1

return t

def main():

m1 = create(4,9,7)

inpm1 = inputmatrix(m1)

print("3D array of size 4 X 9 X 7=")

print()

print(inputmatrix(inpm1))

print()

print("2D arrays of size 9 X 7.")

print()

print(input2Dmatrix(inpm1))

print()

print("4 stacked 2D arrays of size 9 X 7.")

print()

print(input2Dmatrix(inpm1) \* 4)

main()