import random

def prodotto\_vettore\_scalare(vettore,scalare):

ret = []

for valore in vettore:

ret.append(valore \* scalare)

return ret

def somma\_vettori(vettore\_1,vettore\_2):

ret = []

for i in range(len(vettore\_1)):

ret.append(vettore\_1[i]+vettore\_2[i])

return ret

def leggi\_lista\_int\_lunghezza(n):

ret = []

for i in range(n):

v = int(input("Inserisci l'elemento di indice "+str(i)+":"))

ret.append(v)

return ret

def leggi\_lista\_int():

n = int(input('Lunghezza:'))

return leggi\_lista\_int\_lunghezza(n)

def leggi\_lista\_float\_lunghezza(n):

ret = []

for i in range(n):

v = float(input("Inserisci l'elemento di indice "+str(i)+":"))

ret.append(v)

return ret

def leggi\_lista\_float():

n = int(input('Lunghezza:'))

return leggi\_lista\_float\_lunghezza(n)

def prodotto\_scalare(vettore\_1,vettore\_2):

s = 0

for i in range(len(vettore\_1)):

s += vettore\_1[i] \* vettore\_2[i]

return s

def e\_quadrata(M):

return len(M) == len(M[0])

def e\_nulla(M):

m = len(M)

n = len(M[0])

for i in range(m):

for j in range(n):

if M[i][j] != 0:

return False

return True

def e\_diagonale(M):

if not e\_quadrata(M):

return False

dim = len(M)

for i in range(dim):

for j in range(dim):

if i != j and M[i][j] != 0:

return False

return True

def e\_identita(M):

if not e\_quadrata(M):

return False

dim = len(M)

for i in range(dim):

for j in range(dim):

if i == j and M[i][j] != 1 or i != j and M[i][j] != 0:

return False

return True

def crea\_matrice(num\_righe, num\_colonne, valore):

ret = []

for i in range(num\_righe):

riga = [valore] \* num\_colonne

ret.append(riga)

return ret

def stampa\_matrice(M):

for riga in M:

print(riga)

def crea\_matrice\_nulla(num\_righe,num\_colonne):

return crea\_matrice(num\_righe,num\_colonne,0)

def crea\_matrice\_identita(dim):

ret = crea\_matrice\_nulla(dim,dim)

for k in range(dim):

ret[k][k] = 1

return ret

def somma\_diagonale\_principale(M):

s = 0

dim = len(M)

for k in range(dim):

s += M[k][k]

return s

def somma\_diagonale\_secondaria(M):

s = 0

dim = len(M)

for i in range(dim):

s += M[i][dim-1-i]

return s

def elimina\_riga(M,i):

del M[i]

def elimina\_colonna(M,j):

for riga in M:

del riga[j]

def scambia\_righe(M,i1,i2):

t = M[i1]

M[i1] = M[i2]

M[i2] = t

def scambia\_colonne(M,j1,j2):

for riga in M:

t = riga[j1]

riga[j1] = riga[j2]

riga[j2] = t

def prodotto\_matrice\_vettore(M,V):

ret = []

for riga in M:

ps = prodotto\_scalare(riga,V)

ret.append(ps)

return ret

def prodotto\_matrice\_scalare(M,s):

num\_righe = len(M)

num\_colonne = len(M[0])

ret = crea\_matrice\_nulla(num\_righe,num\_colonne)

for i in range(num\_righe):

for j in range(num\_colonne):

ret[i][j] = M[i][j] \* s

return ret

def trasposta(M):

num\_righe\_M = len(M)

num\_colonne\_M = len(M[0])

ret = crea\_matrice\_nulla(num\_colonne\_M,num\_righe\_M)

for i in range(num\_righe\_M):

for j in range(num\_colonne\_M):

ret[j][i] = M[i][j]

return ret

def copia\_matrice(M):

ret = []

for riga in M:

ret.append(riga[:])

return ret

def e\_simmetrica(M):

if not e\_quadrata(M):

return False

dim = len(M)

for i in range(dim-1):

for j in range(i+1,dim):

if M[i][j] != M[j][i]:

return False

return True

def crea\_matrice\_random(num\_righe,num\_colonne,valore\_max):

ret = []

for i in range(num\_righe):

riga = []

for j in range(num\_colonne):

riga.append(random.randint(0,valore\_max))

ret.append(riga)

return ret

def prodotto\_matrici(A,B):

m = len(A)

n = len(A[0])

p = len(B[0])

R = crea\_matrice\_nulla(m,p)

for i in range(m):

for j in range(p):

valore = 0

for k in range(n):

valore += A[i][k] \* B[k][j]

R[i][j] = valore

return R

def leggi\_matrice\_int\_dimensioni(nr,nc):

matrice = []

for i in range(nr):

print('-- Riga '+str(i)+' --')

riga = leggi\_lista\_int\_lunghezza(nc)

matrice.append(riga)

return matrice

def leggi\_matrice\_int():

r = int(input('Numero righe = '))

c = int(input('Numero colonne = '))

return leggi\_matrice\_int\_dimensioni(r,c)

def leggi\_matrice\_float\_dimensioni(nr,nc):

matrice = []

for i in range(nr):

print('-- Riga '+str(i)+' --')

riga = leggi\_lista\_float\_lunghezza(nc)

matrice.append(riga)

return matrice

def leggi\_matrice\_float():

r = int(input('Numero righe = '))

c = int(input('Numero colonne = '))

return leggi\_matrice\_float\_dimensioni(r,c)

def stampa\_matrice\_incolonnata(matrice,n):

'''Incolonna numeri fino a n cifre'''

for riga in matrice:

for elemento in riga:

print(str(elemento).rjust(n+1), end='')

print()