Algoritmos T.P. N° 1 - Parte 1 - Juan Cruz Ambrosini

Ejercicio 1:

from algo1 import \*

import random

import math

dimensionVector = 10

vector = Array(dimensionVector, 0)

for i in range (0, dimensionVector):

    vector[i] = random.randint(-100,100)

print(vector)

mayorValor = -1

for i in range (0, dimensionVector):

    if (mayorValor < abs(vector[i])):

        mayorValor = abs(vector[i])

print("El mayor valor es: ", mayorValor)



Ejercicio 2:

from algo1 import \*

import random

import math

suma = 0

dimension\_vector1 = 10

vector1 = Array(dimension\_vector1, 0)

dimension\_vector2 = 10

vector2 = Array(dimension\_vector2, 0)

for i in range (0, dimension\_vector1):

    vector1[i] = random.randint(0,50)

for i in range (0, dimension\_vector2):

    vector2[i] = random.randint(0,50)

print("Vector 1")

print(vector1)

print("Vector 2")

print(vector2)

if (dimension\_vector1 == dimension\_vector2):

    vectorResultado = Array(dimension\_vector1, 0)

    for i in range (0, dimension\_vector1):

        vectorResultado[i] = vector1[i] + vector2[i]

    print("Vector resultado:")

    print(vectorResultado)

    for i in range (0, dimension\_vector1):

        suma += vectorResultado[i]^2

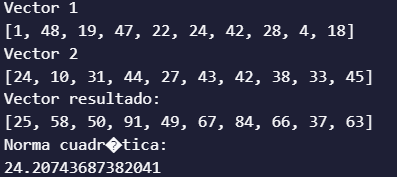
    norma\_cuadratica = math.sqrt(suma)

    print("Norma cuadrática: ")

    print(norma\_cuadratica)

else:

    print("Los vectores no tienen el mismo tamaño y no pueden ser sumados.")



Ejercicio 3:

from algo1 import \*

import random

*#--------------------------------------------------------------------------------------------------------*

dimension\_vector = 4

filas = 4

columnas = 5

suma\_parcial = 0

vector = Array(dimension\_vector, 0)

matriz = Array(filas, Array(columnas, 0))

vector\_resultado = Array(columnas, 0)

*#--------------------------------------------------------------------------------------------------------*

def mostrar\_matriz(matriz, filas):

    for i in range (0, filas):

      print(matriz[i])

*#--------------------------------------------------------------------------------------------------------*

for i in range (0, dimension\_vector):

    vector[i] = random.randint(-20, 20)

print("----------------------------------------------------------")

print (vector)

print("----------------------------------------------------------")

for i in range (0, filas):

    for j in range (0, columnas):

        matriz[i][j] = random.randint(-20, 20)

print("----------------------------------------------------------")

mostrar\_matriz(matriz, filas)

print("----------------------------------------------------------")

if (dimension\_vector == filas):

    for i in range (0, columnas):

        suma\_parcial = 0

        for j in range (0, dimension\_vector):

            suma\_parcial += matriz[j][i] \* vector[j]

        vector\_resultado[i] = suma\_parcial

    print("----------------------------------------------------------")

    print(vector\_resultado)

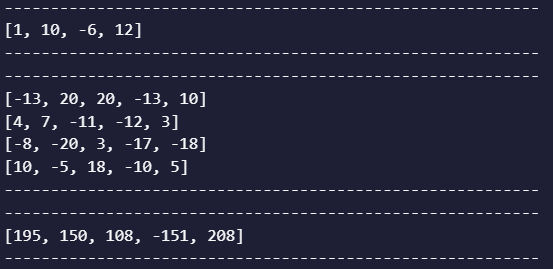
    print("----------------------------------------------------------")

else:

    print("----------------------------------------------------------")

    print("Dimensiones incorrectas")

    print("----------------------------------------------------------")



Ejercicio 4:

from algo1 import \*

import random

filas = 5

columnas = 5

def crear\_matriz (filas, columnas):

    matriz = Array(filas, Array(columnas, 0))

    for i in range (0, filas):

        for j in range (0, columnas):

            matriz[i][j] = random.randint(-20,20)

    return matriz

def mostrar\_matriz(matriz):

    for i in range (0, filas):

      print(matriz[i])

def sumar\_matrices(matriz\_a, matriz\_b, filas, columnas):

    matriz\_resultado = Array(filas, Array(columnas, 0))

    for i in range(0, filas):

        for j in range(0, columnas):

            matriz\_resultado[i][j] = matriz\_a[i][j] + matriz\_b[i][j]

    return matriz\_resultado

print("----------------------------------------------------------")

print("Matriz A")

print("----------------------------------------------------------")

matriz\_a = crear\_matriz(filas, columnas)

mostrar\_matriz(matriz\_a)

print("----------------------------------------------------------")

print("Matriz B")

print("----------------------------------------------------------")

matriz\_b = crear\_matriz(filas, columnas)

mostrar\_matriz(matriz\_b)

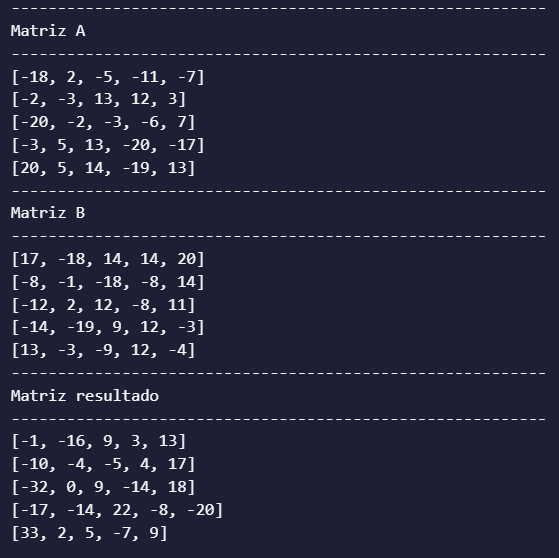
print("----------------------------------------------------------")

print("Matriz resultado")

print("----------------------------------------------------------")

matriz\_resultado = sumar\_matrices(matriz\_a, matriz\_b, filas, columnas)

mostrar\_matriz(matriz\_resultado)



Ejercicio 5:

from algo1 import \*

import random

dimension = 3

matriz = Array(dimension, Array(dimension, 0))

determinante = 1

matriz\_no\_triang\_sup = Array(dimension, Array(dimension, 0))

*#--------------------------------------------------------------------------------------------------------*

def crear\_matriz (filas, columnas):

    matriz = Array(filas, Array(columnas, 0))

    for i in range (0, filas):

        for j in range (0, columnas):

            matriz[i][j] = random.randint(-20,20)

    return matriz

def validar\_si\_es\_triang\_sup (matriz, dimension):

    es\_triang\_sup = True

    for i in range (0, dimension):

        for j in range(0, dimension):

            if (j > i and matriz[j][i] != 0):

                es\_triang\_sup = False

    return es\_triang\_sup

def mostrar\_matriz(matriz):

    for i in range (0, dimension):

      print(matriz[i])

*#--------------------------------------------------------------------------------------------------------*

for i in range(0, dimension):

    for j in range(0, dimension):

        if (j > i):

            matriz[j][i] = 0

        else:

            matriz[j][i] = random.randint(-20,20)

*# matriz = crear\_matriz(dimension, dimension)          #Esta línea esta a modo de verificicación con el fin de evaluar como se comportaría el algoritmo con una matriz no ts*

mostrar\_matriz(matriz)

es\_triang\_sup = validar\_si\_es\_triang\_sup(matriz, dimension)

if(es\_triang\_sup):

    print("La matriz si es triangular superior.")

    for i in range(0, dimension):

        for j in range (0, dimension):

            if i == j:

                determinante = determinante \* matriz[i][j]

    print("Su detereminante es igual a: ", determinante)

else:

    print("La matriz no es triangular superior y su determinante no será calculado.")

