

In []:

```
### Parcial II Fund. Prog.
### FUnciones en Python (programación estructurada)
### Nicolás Rendón Arias, Manuel Alejandro Bermúdez, Oscar Julián Toro
```

In [27]:

```
##### Inicio parcial #####
'''
1)
Genere dos matriz de tamaño 5, almacene valores aleatorios de 1 a 100 y calcule:
a. Suma de todos los elementos de la primer matriz
b. Media de la matriz
c. Desviación estándar
d. Mediana
e. Mayor valor
f. Menor valor
g. Suma las dos matrices
h. Multiplicación de las dos matrices
'''
#####(1)#####
import random #importa la librería random
import numpy as np #importa la librería numpy con abreviación
lisVal=[] #declara la lista 1
lisVal2=[] #declara la lista 2

#####(A)#####
#####(Primera matriz)#####

for x in range (25): #para X en el rango de 25...
    val=random.randrange(1,100) #valor toma un número aleatorio entre 1 y 100
    lisVal.append(val) #la lista 1 recibe el valor

acumSum=0 #declara un acumulador de la suma
for x in range (len(lisVal)): #para X en el rango de la cantidad de elementos en la lista 1...
    acumSum=acumSum+lisVal[x] #se suma el elemento en posición X al acumulado
print(np.array(lisVal).reshape(5,5))
print('La suma es: ', acumSum) #se imprime la suma acumulada

#####(Fin A)#####

[[20  4 53 69 67]
 [86 80 79 89 96]
 [14 53 26 63 55]
 [94 26 64 20 25]
 [17 23 75  9 54]]
La suma es: 1261
```

In [26]:

```
#####(B y C)#####
import random #importa librería random
import math #importa librería matemática
import numpy #importa librería numpy
listaDatos=[] #declara lista de datos
restar=[] #declara lista para la resta
elevar=[] #declara lista para elevar
def sumatoria(datos): #define funcion sumatoria
    sumatoria = float(sum(datos)) #declara variable sumatoria con suma de datos como float
    return sumatoria #devuelve sumatoria
# * Paso 2 *
def media(datos): #define funcion 'media'
    n = len(datos) #define variable con longitud del parámetro
    mediana = sumatoria(datos) / n #operación para la media
    return round(mediana,2) #devuelve el resultado con dos decilames
# * Paso 3 *
def restar_media_datos(datos): #define función para restar media de datos

    mediana = media(datos) #declara variable mediana
    print('Media: ', mediana) #imprime la media
```

```

    for i in datos: #para una variable en 'datos'...
        op = i - mediana #opera con la mediana...
        restar.append(op) #restar recibe a la variable

# * Paso 4 *
def elevar_cuadrado(datos): #define funcion elevar al cuadrado

    for i in datos: #para una variable en 'datos'...
        op = pow(i, 2) #eleva la variable al cuadrado
        elevar.append(op) #elevar recibe el resultado

# * Paso 6 *

def raiz_datos(): #define la funcion raiz de datos...
    desviacion = math.sqrt(media(elevar)) #le saca raiz cuadrada a la variable en parentesis...
    return round(desviacion,2) #devuelve el resultado con dos decimales

# * Inicio *
def main(): #define funcion principal...

    repetir = int(input('Cuantos valores quieres ingresar? : ')) #pide al usuario cantidad de
    valores a ingresar...

    for i in range(repetir): #para i en el rango de la cantidad de valores dada por el usuario...
        number = random.randrange(25) #elige un valor aleatorio
        listaDatos.append(number) #agregar el valor a la lista de datos
    print(numpy.array(listaDatos).reshape(5,5)) #lo transforma a matriz con dimensiones 5x5
    suma = sumatoria(listaDatos) #define variable suma
    print('Sumatoria: ', suma) #imprime la suma

    restar_media_datos(listaDatos) #llama a la funcion

    elevar_cuadrado(restar) #llama a la funcion

    mediana = media(elevar) #define variable mediana
    print('Varianza: ', mediana) #imprime la varianza

    desviacion = raiz_datos() #define variable desviacion
    print('Desviación Estándar: ', desviacion) #imprime desviación estandar
    print('\n') #imprime salto de línea

main() #llamado a la función
#===== (Fin B y C) =====

```

```

[98, 14, 17, 25, 89, 27, 7, 87, 19, 94, 20, 36, 77, 33, 91, 55, 79, 62, 61, 80, 76, 36, 72, 9, 22
]
La suma es: 1286
Cuantos valores quieres ingresar? : 25
[[10 15 17  5 22]
 [22  5 20  4 19]
 [13  7  4 22 19]
 [ 9  9 22 12  6]
 [ 2 22 18  7  9]]
Sumatoria: 320.0
Media: 12.8
Varianza: 46.4
Desviación Estándar: 6.81

```

In [3]:

```

aux=0 #se declara la variable auxiliar
for x in range(1,len(lisVal)): #para X en el rango de 1 hasta la longitud de la lista 1...
    for y in range(0,len(lisVal)-x): #para Y en el rango de 0 hasta la longitud de la lista 1 -
    X...
        if lisVal[y]>lisVal[y+1]: #si el elemento actual es mayor al que le sigue...
            aux=lisVal[y+1] #auxiliar toma el valor del elemento que le sigue...
            lisVal[y+1]=lisVal[y] #el elemento Y+1 toma el elemento en posición Y
            lisVal[y]=aux #el elemento en posición Y toma el valor del auxiliar...
medMat=lisVal[12] #variable para la mediana de la matriz
print(lisVal) #imprime la lista 1

#===== (D) =====
print('La mediana es: ', medMat) #imprime la mediana

```

```

#===== (E) =====
print('El mayor elemento es: ', lisVal[24])    #imprime el mayor elemento de la matriz
#===== (F) =====
print('El menor elemento es: ', lisVal[0])    #imprime el menor elemento de la matriz

print(np.array(lisVal).reshape(5,5))    #convierte la lista en la matriz solicitada con datos aleato
rios.

#===== (Segunda matriz) =====

for x in range (25):    #para X en el rango de 25...
    val2=random.randrange(1,100)    #val2 toma un valor aleatorio entre 1 y 100
    lisVal2.append(val2)    #la lista2 recibe a val2

acumSum2=0    #variable que acumula la suma
for x in range (len(lisVal2)):    #para x en el rango de la longitud de la lista 2...
    acumSum2=acumSum2+lisVal2[x]    #se suma el elemento en posición X al acumulado
print('La suma es: ', acumSum2)    #imprime la suma

aux2=0    #se declara la variable auxiliar2
for x in range (1,len(lisVal2)):    #para X en el rango de 1 a la longitud de la lista2...
    for y in range (0,len(lisVal2)-x):    #para Y en el rango de 0 hasta longitud de lista2 - X...
        if lisVal2[y]>lisVal2[y+1]:    #si el elemento actual es mayor al que le sigue...
            aux2=lisVal2[y+1]    #la auxiliar toma el valor del elemento en posición Y+1
            lisVal2[y+1]=lisVal2[y]    #el elemento Y+1 toma el valor del actual
            lisVal2[y]=aux2    #el elemento en posición Y toma el valor del auxiliar
medMat2=lisVal2[12]    #Declara la mediana de la lista ordenada
print(lisVal2)    #imprime la lista2
print('El mayor elemento es: ', lisVal2[24])    #imprime el mayor elemento de la lista2
print('El menor elemento es: ', lisVal2[0])    #imprime el menor elemento de la lista2
print('La mediana es: ', medMat2)    #imprime la mediana de la lista2

print(np.array(lisVal2).reshape(5,5))    #convierte la lista en la matriz solicitada con datos aleat
orios.

```

```

[2, 9, 12, 13, 18, 20, 21, 24, 28, 31, 34, 38, 44, 49, 50, 51, 53, 53, 56, 64, 70, 74, 82, 85, 97
]
La mediana es:  44
El mayor elemento es:  97
El menor elemento es:  2
[[ 2  9 12 13 18]
 [20 21 24 28 31]
 [34 38 44 49 50]
 [51 53 53 56 64]
 [70 74 82 85 97]]
La suma es:  1402
[4, 8, 9, 14, 15, 29, 31, 38, 44, 46, 49, 49, 60, 73, 74, 75, 80, 82, 82, 84, 85, 90, 91, 93, 97]
El mayor elemento es:  97
El menor elemento es:  4
La mediana es:  60
[[ 4  8  9 14 15]
 [29 31 38 44 46]
 [49 49 60 73 74]
 [75 80 82 82 84]
 [85 90 91 93 97]]

```

In [4]:

```

# ===== (G) =====
# suma de dos matrices aleatorias
from numpy import *    #importa librería numpy
import numpy as np    #abrevia librería numpy
import random    #importa librería random

matriz1 = []    #declara lista para matriz1
matriz2 = []    #declara lista para matriz2
matrizsum = []    #declara lista para suma de las matrices
numero_filas = int(input("ingrese el numero de filas para la primer matriz "))    #pide las filas d
e la primera matriz
numero_columnas = int(input("ingrese el numero de columnas para la primer matriz "))    #pide las c
olumnas de la primera matriz

def func_valran():    #define la función
    for conCic in range(25):    #para el contador del ciclo en el rango 25...

```

```

        valor = random.randrange(20)    #se asigna un valor aleatorio
        return valor    #retorna el valor

for i in range(numero_filas):    #para la variable en el rango del numero de filas...
    matriz1.append([])    #matriz1 recibe valor
    matrizsum.append([])    #matrizsuma recibe valor
    for j in range(numero_columnas):    #para j en el rango del numero de columnas...
        matriz1[i].append(func_valran())    #la matriz en posición i recibe valor...
        matrizsum[i].append(0)    #la matriz suma en posición i recibe 0

for m in range(numero_filas):    #para m en el rango del numero de filas...
    matriz2.append([])    #matriz2 recibe valor...
    for n in range(numero_columnas):    #para n en rango de las columnas...
        matriz2[m].append(func_valran())    #matriz2 en posición m recibe valor...
print(array(matriz1))    #imprime como array matriz1
print("")    #imprime separador
print(array(matriz2))    #imprime como array matriz2

for conI in range(numero_filas):    # para el contador en el rango del numero de filas....
    for conJ in range(numero_columnas):    #para el contador en el rango del numero de columnas...
        matrizsum[conI][conJ] = (matriz1[conI][conJ] + matriz2[conI][conJ])    #opera la suma de las
matrices

print("La suma de las matrices es: ")    #imprime el aviso...
print(array(matrizsum))    #imprime la suma de las matrices como un array

```

```

ingrese el numero de filas para la primer matriz 5
ingrese el numero de columnas para la primer matriz 5
[[ 2 16  3  2  4]
 [16 17  1 10  9]
 [ 4  7 16  9 12]
 [11 18 15 11 18]
 [12 10  6  8 15]]

```

```

[[11  0  8 19 13]
 [17  7  8  2  0]
 [ 1  4  5  5  9]
 [19 12 16  3 16]
 [12  5  1 17  6]]

```

La suma de las matrices es:

```

[[13 16 11 21 17]
 [33 24  9 12  9]
 [ 5 11 21 14 21]
 [30 30 31 14 34]
 [24 15  7 25 21]]

```

In [5]:

```

# ===== (H) =====
from numpy import *    #importa la librería numpy
import numpy as np    #abrevia la importación como np
import random    #importa la librería random

matriz1 = []    #declara lista para matriz1
matriz2 = []    #declara lista para matriz2
numero_filas1 = int(input("ingrese el numero de filas para la primer matriz "))    #pide el ingreso
de cantidad filas
numero_columnas1 = int(input("ingrese el numero de columnas para la primer matriz "))    #pide el i
ngreso de cantidad columnas

def func_valran():    #define la función
    for conCic in range(25):    #para el contador del ciclo en el rango de 25...
        valor = random.randrange(20)    #se obtiene un valor aleatorio...
        return valor    #devuelve el valor

for i in range(numero_filas1):    #para i en el rango del numero de filas
    matriz1.append([])    #la matriz recibe un valor...
    for j in range(numero_columnas1):    #para j en el rango del numero de columnas...
        matriz1[i].append(func_valran())    #la matriz en posición i recibe valor...

numero_filas2 = int(input("ingrese el numero de filas para la segunda matriz "))    #pide numero de
filas matriz2

```

```

numero_columnas2 = int(input("ingrese el numero de columnas para la segunda matriz ")) #pide
numero de columnas matriz2
for m in range(numero_filas2): #para m en el rango del numero de filas...
    matriz2.append([]) #la matriz2 recibe valor...
    for n in range(numero_columnas2): #para n en rango del numero de columnas...
        matriz2[m].append(func_valran()) #matriz2 recibe valor...
print(array(matriz1)) #imprime matriz1 como array
print("") #imprime separador
print(array(matriz2)) #imprime matriz2 como array

print("La multiplicacion de las 2 matrices es: ") #imprime un aviso...
print(np.dot(matriz1, matriz2)) #imprime la multiplicación de las matrices con apoyo de librería
numpy as np

```

ingrese el numero de filas para la primer matriz 5
 ingrese el numero de columnas para la primer matriz 4
 ingrese el numero de filas para la segunda matriz 4
 ingrese el numero de columnas para la segunda matriz 5

```

[[ 4 12 11  7]
 [18  7 10  2]
 [14  7 14 17]
 [ 1  4  5  6]
 [ 5  7 16 15]]

```

```

[[ 8 17 19 15 14]
 [ 3 10 13 11  8]
 [14  6 19 17  5]
 [ 7 12  8  6 10]]

```

La multiplicacion de las 2 matrices es:

```

[[271 338 497 421 277]
 [319 460 639 529 378]
 [448 596 759 627 492]
 [132 159 214 180 131]
 [390 431 610 514 356]]

```

In [24]:

```

'''
2)
En matemáticas, la sucesión o serie de Fibonacci es la siguiente sucesión infinita
de números naturales:
Desarrolle una funcion que pida una posición e imprima la serie gibbonacci hasta ese
termino.
'''

#===== Inicio sucesión de Fibonacci =====

def fib_i(n): #define la función para fib
    a, b = 0, 1 # variables a y b toman valores de 0 y 1
    while n > 0: #mientras n sea mayor a 0...
        a, b = b, a + b #opera con sumas la sucesion en a y b
        n -= 1 # n va disminuyendo
    return a #devuelve a
num=int(input(" Ingrese el numero final de la sucesion: ")) #pide el límite de la sucesion
for i in range(num+1): # para i en el rango del número+1...
    print(fib_i(i)) #llamado a la función y la imprime

#===== Fin sucesión de Fibonacci =====

```

Ingrese el numero final de la sucesion: 10

```

0
1
1
2
3
5
8
13
21
34
55

```

In [12]:

```
'''
2.1)
Pida un término para K y Calcule el valor de la sumatoria
'''

# ===== Inicio serie geométrica alternada =====

import math    #importa la librería matemática

k=int(input('Deme un término para K: '))    #pide el valor para K...

def func_serGeom(k):    #define la función de la sucesión...
    conRep=100    #establece contador de repeticiones...
    acuSum=0    #declara un acumulador para la suma
    for x in range (conRep):    #para X en el rango del contador de repeticiones...
        sum=((math.pow(-1,k))*(math.pow(2/3,k-1)))    #opera la fórmula de la sumatoria
        acuSum=acuSum+sum    #el resultado se va guardando en el acumulado
        k=k+1    #la variable K va incrementando con el ciclo...

    return acuSum    # devuelve el acumulado

resultFnal=func_serGeom(k)    #variable con llamado a la función como parámetro K
print('La serie geométrica de ', k, 'es: ')    #imprime un aviso...
print(round(resultFnal,3))    #imprime el resultado y lo redondea a 3 decimales.

# ===== Fin serie geométrica alternada =====
```

Deme un término para K: 5
 La serie geométrica de 5 es:
 -0.119

In [15]:

```
'''
3)
Pida un término para K y Calcule el valor de la sumatoria
'''

# ===== Inicio serie p alternada =====

import math    #importa librería matemática

k=int(input('Deme un término para K: '))    #pide el valor de K al usuario...

def func_serAlt(k):    #define la función para serie p alternada
    conRep=100    #establece contador de repeticiones
    acuSum=0    #declara un acumulador de suma
    for x in range (conRep):    #para X en el rango del contador de repeticiones...
        sum=((math.pow(-1,k-1))*(1/(math.pow(k,1/3))))    #opera la fórmula de la sumatoria...
        acuSum=acuSum+sum    #adiciona el resultado al acumulado...
        k=k+1    #incremento de la variable K con el ciclo

    return acuSum    #devuelve el acumulado

resultFnal=func_serAlt(k)    #establece variable con llamado a la función y parámetro K
print('La serie P alternada de: ', k, 'es: ')    #imprime un aviso
print(round(resultFnal,3))    #imprime el resultado y lo redondea con 3 decimales.

# ===== Fin serie p alternada =====
```

Deme un término para K: 7
 La serie P alternada de: 7 es:
 0.162

In [21]:

```
'''
4)
Genera una lista de 10 posiciones con datos aleatorios y desarrolle las funciones que:
• Sume los elementos del vector
• Calcule la suma de los pares y de impares
• Calcule la suma de los elementos de las posiciones impares
• Que pida una posición y que inserte un valor en dicha posición
• Ordene el vector
```

• Con el vector ordenado que lea un numero por teclado y lo inserte en la posición correcta
'''

```
# ===== (4) =====  
import random #importa la librería random  
  
lista = [] #define la lista  
  
for conCic in range(10): #para el contador del ciclo en el rango de 10...  
    lista.append(random.randrange(1, 20)) #la lista recibe un valor aleatorio entre 1 y 20  
  
print(lista) #imprime la lista  
  
# ===== (4.1) =====  
def func_suma(lista): #define la función de suma  
    suma = 0 #variable para acumulado de suma  
    for conCic in range(len(lista)): #para el contador de ciclo en el rango de la longitud de la lista...  
        suma = suma + lista[conCic] #el elemento en posición del contador se añade al acumulado  
        print("La suma de todos los valores de la lista es: ", suma) #imprime un aviso y la suma  
  
func_suma(lista) #llamado a la función  
  
# ===== (4.2) =====  
def func_sumaParImpar(lista): #define función para pares e impares  
    sumapar = 0 #variable para suma de pares  
    sumaimpar = 0 #variable para suma de impares  
    for conCic in range(len(lista)): #para el contador de ciclo en el rango de la longitud de la lista...  
        if lista[conCic] % 2 == 0: #si el residuo de la división/2 del elemento actual es igual a cero...  
            sumapar = sumapar + lista[conCic] #lo añade al acumulado de suma pares  
        else: #sino...  
            sumaimpar = sumaimpar + lista[conCic] #lo añade al acumulado de suma impares  
        print("La suma de los valores pares es: ", sumapar) #imprime la suma pares  
        print("La suma de los valores impares es: ", sumaimpar) #imprime la suma impares  
  
func_sumaParImpar(lista) #llamado a la función  
  
# ===== (4.3) =====  
def func_sumaPosImpar(lista): #define función para suma de pos impares  
    sumapos = 0 #variable para acumulado de suma  
    for conCic in range(len(lista)): #para el contador del ciclo en el rango de la longitud de la lista...  
        if lista.index(lista[conCic]) % 2 != 0: #con el index devuelve la pos del elemento en cuestión...  
            #si el residuo de su división/2 es diferente de cero...  
            sumapos = sumapos + lista[lista.index(lista[conCic])] #lo añade al acumulado  
        print("la suma de las posiciones impares de la lista es: ", sumapos) #imprime el acumulado  
  
func_sumaPosImpar(lista) #llamado a la función  
print(" ")  
# ===== (4.4) =====  
def func_insertar(lista): #define la función insertar  
    valor = int(input("Ingrese la posición: ")) #se le pide un valor al usuario...  
    nuevovalor = int(input("Ingrese el valor a insertar: ")) #se pide un valor nuevo a insertar..  
    .  
    lista[valor] = nuevovalor #se hace el intercambio  
    print("Lista con el valor modificado: ", lista) #imprime el vector modificado  
  
func_insertar(lista) #llamado a la función  
print(" ")  
# ===== (4.5) =====  
def func_ordenarlista(lista): #define función para ordenar lista  
    for conCic in range(len(lista) - 1, 0, -1): #para el contador del ciclo en el rango de 1-0-1 longitud list...  
        for i in range(conCic): #para i en el rango del contador...  
            if lista[i] > lista[i + 1]: #si el elemento actual es mayor al que le sigue...  
                temp = lista[i] #guarda el valor del actual en variable temporal...  
                lista[i] = lista[i + 1] #valor actual toma el valor del que le sigue  
                lista[i + 1] = temp #el valor en pos i+1 toma el valor del temporal  
        print("Lista ordenada: ", lista) #imprime lista ordenada
```

```

func_ordenarlista(lista)  #llamado a la función con la lista como parámetro
print(" ")
# ===== (4.6) =====
def func_insertordenada(lista):  #define funcion para insertar en pos correcta...
    valor = int(input("Ingrese un valor: "))  #se pide al user el ingreso del valor
    lista.append(valor)  #la lista recibe al valor
    for conCic in range(len(lista) - 1, 0, -1):  #para el contador del ciclo en el rango de 1-0-1
        longitud list...
        for i in range(conCic):  #para i en el rango del contador...
            if lista[i] > lista[i + 1]:  #si el elemento actual es mayor al que le sigue...
                temp = lista[i]  #guarda el valor del actual en variable temporal...
                lista[i] = lista[i + 1]  #valor actual toma el valor del que le sigue
                lista[i + 1] = temp  #el valor en pos i+1 toma el valor del temporal
    print("Lista con el nuevo valor ordenado", lista)  #imprime lista modificada ordenada

func_insertordenada(lista)  #llamado a la función con lista como parámetro...

#===== Fin parcial =====

```

```

[9, 15, 4, 2, 10, 1, 5, 16, 11, 6]
La suma de todos los valores de la lista es: 79
La suma de los valores pares es: 38
La suma de los valores impares es: 41
la suma de las posiciones impares de la lista es: 40

```

```

Ingrese la posicion: 5
Ingrese el valor a insertar: 78
Lista con el valor modificado: [9, 15, 4, 2, 10, 78, 5, 16, 11, 6]

Lista ordenada: [2, 4, 5, 6, 9, 10, 11, 15, 16, 78]

```

```

Ingrese un valor: 25
Lista con el nuevo valor ordenado [2, 4, 5, 6, 9, 10, 11, 15, 16, 25, 78]

```

In []: