

# Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

Profesor:	Adrian Ulises Mercado
Asignatura:	Estructura de datos y Algoritmos I
Grupo:	13
No de Práctica(s):	Práctica 10
Integrante(s):	Escobar Ventura Jesus Emiliano
No. de Equipo de cómputo empleado:	
No. de Lista o Brigada:	Brigada 5
Semestre:	2020-2
Fecha de entrega:	<b>07</b> .06.2020
Observaciones:	
CALIFICACIÓN:	

# INTRODUCCIÓN

Conoceremos las estructuras selectivas como if, if-else y elif, ciclos while, for, también desarrollaremos pilas, bibliotecas, y generamos una gráfica con la biblioteca maplotlib

### **DESARROLLO**

### Estructuras selectivas

En este programa se busca encontrar el número mayor de 3 números, los cuales van a ser ingresados desde línea de comandos, con la ayuda de operadores lógicos podemos ver si un número es mayor que otro, si no fuera el caso se pone elif que sería equivalente a switch en c.

Para leer los números que se ingresan desde línea de comando se utiliza input

```
def numeroMayor(a, b, c):
    if a > b and a > c:
        print("El numero mayor es {}".format(a))
    elif (b > c and b > a):
        print ("El numero mayor es {}" .format(b))
    else:
        print("El numero mayor es {}" .format(c))

if __name__ == "__main__":
    a= int(input())
    b= int(input())
    c= int(input())
    numeroMayor(a,b,c)
```

# Estructuras de repetición

# For y Diccionarios

En esta función se utiliza las estructuras de repetición para imprimir listas, bibliotecas y arreglos.

En donde x toma cada valor de la lista y lo imprime hasta el último elemento. El función range sirve para generar una lista.

Para imprimir un diccionario, se usa las palabras key y value, donde key es el primer elmento, seguido de (:) y un valor (value)

```
def forlist():
    for x in [1, 2, 3, 4, 5]:
        print(x)

    for x in ["uno", "dos", "tres", "cuatro", "cinco"]:
        print(x)
```

```
def forrange():
#for para diccionarios
def fordic():
   diccionario = {'manzana': 1, 'pera':3, 'uva':10 }
    for clave in diccionario.keys():
       print(clave)
    for valor in diccionario.values():
        print("el indice {} del elemento {}".format(idx,x))
#else de for
def elsefor():
    for x in range (5):
    else:
```

```
print("la cuenta se termino")

def elsefor2():
    for x in range(5):
        print(x)
        if x==2:
            break
    else:
        print("la cuenta se termino")

if __name__ == "__main__":
    forlist()
    forrange()
```

### while

While es una estructura de repetición, para entenderlo mejor, hicimos el ejemplo del factorial de un número.

```
def factorial(n):
    i = 2
    temp = 1
    while i <= n:
        temp = temp * i
        i = i + 1
    return temp

if __name__ == "__main__":
    a = int(input("Ingresa un numero: "))
    print(factorial(a))</pre>
```

### **Bibliotecas**

Las bibliotecas nos proporcionan funciones con las cuales se puede facilitar los programas, como la biblioteca math que nos proporciona las operación matematicas.

```
#import para importar

import math

x= math.cos(math.pi)
```

```
print(x)

from math import cos,pi #para importa algunas funciones de marh
from math import *
x = cos(pi)
print(x)

#Para imprimir las funciones de math
import math
print(dir(math))

#Con help para ver como funcionan
help(math.log)

import math as x
```

### Gráficas

Con la biblioteca matplotlib podemos graficar funciones en 2d o 3d como la siguiente:

```
#Alead de mathplot.pyplt como plt
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from math import sin

#dato de entrada
x = np.linspace(0, 5, 20)

fig, ax = plt.subplots(facecolor= "w", edgecolor="k")
ax.plot(x, marker="o", color="r", linestyle="None")

ax.grid(True)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.legend(["Y = x"])

plt.title('Puntos')
plt. show()
```

# **EJECUCIÓN**

## Estructuras selectivas

# For y Diccionarios



## while

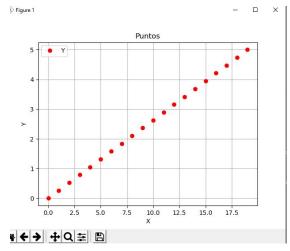
```
SALIDA TERMINAL CONSOLADE DEPURACIÓN PROBLEMAS

PS C:\Users\Emiliano\Desktop\Estructura de datos\Practica 10> & C:\Users\Emiliano\AppData\Local/Programs/Python/Python38-32/python.exe "c:\Users\Emiliano/Desktop\Estructura de datos\Practica 10/factorial.py"
Ingresa un numero: 7
5040

PS C:\Users\Emiliano\Desktop\Estructura de datos\Practica 10>
```

### **Bibliotecas**

### Gráficas



# **CONCLUSIÓN**

Python es un lenguaje interpretado, el cual no necesita ser compilado primero para posteriormente ejecutarlo, python se ejecuta y se para se llega a encontrar algún error.

Por otro lado python es más estricta en la sintaxis, ya que se debe respetar las sangrías.

# **COMENTARIOS**

Me gustaria profundizar mas en el lenguaje, porque sin duda es un lenguaje relativamente sencillo, que puede servir como punto de partida para aprender otros lenguajes.

# **BIBLIOGRAFÍAS**