



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): 10

Integrante(s): Pacheco Salgado Mauricio

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 10

Semestre: 2020-1

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

INTRODUCCION.

Práctica introducción a Python II veremos cómo se manejan las estructuras de datos, en este caso veremos un ejemplo de pila, posteriormente se revisará las declaraciones y usos de bibliotecas, también usaremos funciones repetitivas como es el caso de for y while y veremos cómo se podrá graficar un programa utilizando las librerías matplotlib.

OBJETIVOS

Aplicar las bases del lenguaje de programación Python.

DESARROLLO

programa que utiliza las declaraciones if else, elseif, aplicadas a un programa que ordena números por cantidad(<,>), y se utilizara la función input para pedir datos por terminal y teclado.

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

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

Código practica10.py

- algoritmo para encontrar el número mayor de tres números, los cuales son pedidos al usuario a través de la línea de comandos. Para implementar el algoritmo se hace uso de la sentencia if-elif-else. Esta sentencia es equivalente a la sentencia switch en otros lenguajes.
-

En este programa se hace uso de la sentencia while para obtener el factorial de un número n.

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

if name == " main ":
    a = int(input("ingresa un num"))
    print(fact(a))
```

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

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

En Python, la sentencia for se usa mayormente para recorrer listas. Estas listas pueden ser de números, cadenas de texto, etc. La variable x va tomando el valor de cada elemento dentro de la lista.

La sentencia for x in range() le da valores a x dentro de los parámetros que están dentro de los paréntesis de la función range. El primer parámetro es el número del cuál empieza, el segundo parámetro es el número en el cual termina menos no; y el último parámetro es el incremento.

```
for para rangos
"""
def forrange():
    for x in range(5):
        print(x)

    for y in range(-3,3):
        print(y)

    for z in range(-4, 2, 2):
        print(z)

    for i in range(5, 0, -1):
        print(i)
```

Se utiliza para recorrer diccionarios En la primera sentencia for se recorren todos los elementos del diccionario dándole los valores de value y key a valor y clave respectivamente.

En el siguiente for solamente se recorre el diccionario por las keys; la variable clave es la que va tomando estos valores.

En la tercera sentencia, sólo que en vez de las keys, ahora son los valores.

Por último, se hace uso de la función enumerate para usar índices.

```
"""
def fordic():
    diccionario = {'manzana': 1, 'pera':3, 'uva':10 }
    for clave, valor in diccionario.items():
        print(clave, " = ", valor)

    for clave in diccionario.keys():
        print(clave)

    for valor in diccionario.values():
        print(valor)

    for idx, x in enumerate(diccionario):
        print("el indice {} del elemento {}".format(idx,x))
"""
```

```
else de for
"""

def elsefor():
    for x in range(5):
        print(x)
    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()
```

La sentencia else en el for se ejecuta después de que termina la iteración.

```
import math
from math import *
from math import cos, pi
x= math.cos(math.pi)
x = cos(pi)
print(x)
```

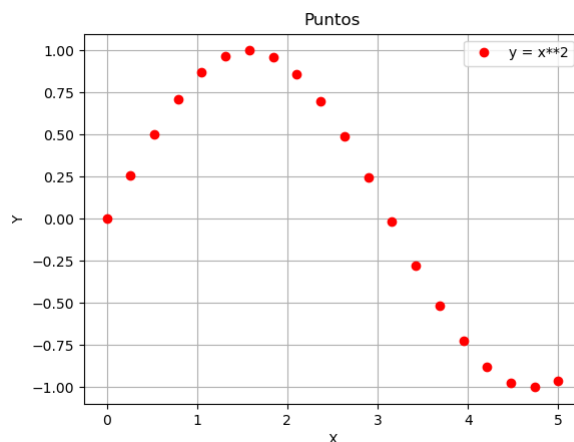
Por medio de la función import se puede importar cualquier biblioteca de Python. En este

ejemplo math. Con la sentencia de la línea 3 se imprimen las funciones contenidas en la biblioteca y con la sentencia en la siguiente línea se imprime la forma en la que se tienen que utilizar las funciones.

```
1 | import math
2 |
3 | print(dir(math))
4 | #uso de la funcion
5 | help(math.log)
6 |
7 | #se le pueden poner apodos a las bibliotecas
8 | x = math.cos(math.pi)
```

```
1 | #Importando las bibliotecas
2 | import matplotlib.pyplot as plt
3 | from mpl_toolkits.mplot3d import Axes3D
4 | from numpy import *
5 |
6 | #Datos de entrada
7 | x = linspace(0, 5, 20) #Generando 10 puntos entre 0 y 5
8 |
9 |
10 | fig, ax = plt.subplots(facecolor='w', edgecolor='k')
11 | ax.plot(x, sin(x), marker="o", color="r", linestyle='None')
12 |
13 | ax.grid(True)
14 | ax.set_xlabel('X') #Etiqueta del eje x
15 | ax.set_ylabel('Y') #Etiqueta del eje y
16 | ax.grid(True)
17 | ax.legend(["y = x**2"])
18 |
19 | plt.title('Puntos')
20 | plt.show()
21 |
22 | fig.savefig("gráfica.png") #Guardando la gráfica
```

Para este código se hace uso de la biblioteca Matplotlib que sirve para hacer gráficas en 2D y 3D y sirve bastante eficiente. En este ejemplo se hace la gráfica de la función seno con algunos generados entre 0 y 5 con ayuda de una función de la biblioteca Numpy.



Conclusiones

Se logro cumplir el objetivo de la practica ya que si se obtuvieron los conocimientos básicos y algo intermedios para realizar códigos en Python a un nivel básico y las bases con las diferentes funciones de ellas