



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Adrian Ulises Mercado Martínez

Asignatura: Estructura de Datos y Algoritmos I.

Grupo: 13

No de Práctica(s): 10

Integrante(s): Nicolás López Daniela

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 36

Semestre: 2020-2

Fecha de entrega: 07-06-2020

Observaciones:

CALIFICACIÓN: _____

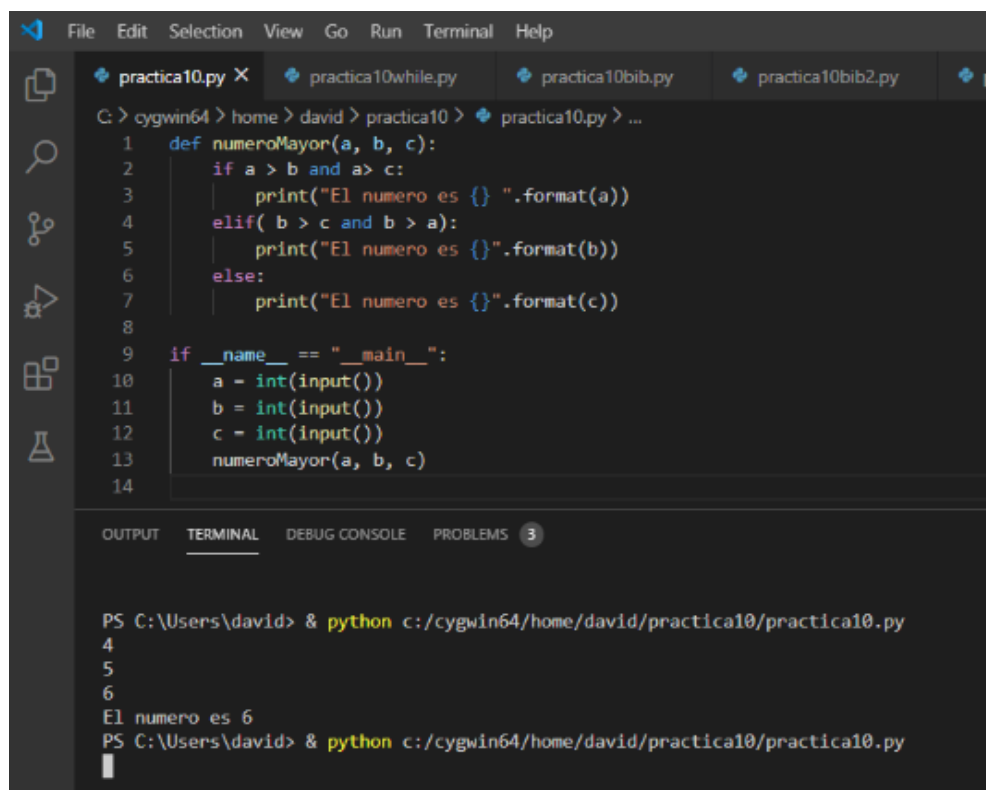
❖ INTRODUCCION.

En esta práctica realice varios programas donde el asunto principal era utilizar nuevos comandos en Python además de utilizar las distintas librerías

❖ DESARROLLO (CON EJERCICIOS)

▪ PROGRAMA 1.

En este primer ejercicio se hace un programa que calcula el número mayor que des, a partir de tres números, esto es por medio de comparaciones en if, elif y else. La diferencia de un if en Python y en C no es mucha sólo que en Python no utilizamos los paréntesis en los argumentos de un if.

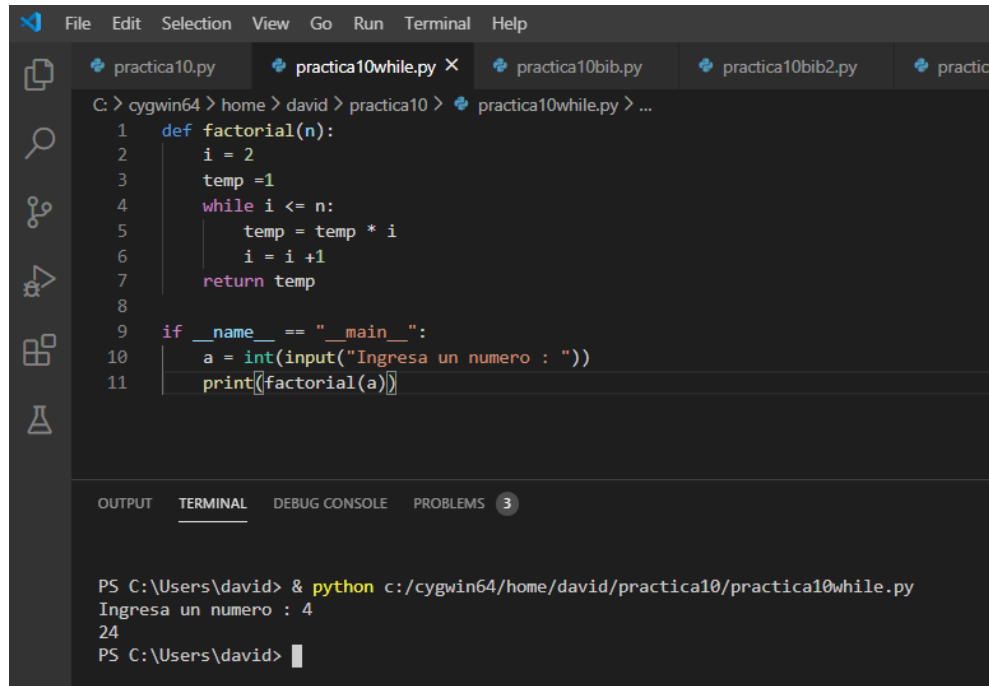


The screenshot shows a Python IDE with a file explorer on the left and a code editor. The code editor displays a Python script named `practica10.py`. The script defines a function `numeroMayor(a, b, c)` that takes three arguments and returns the maximum value using `if`, `elif`, and `else` statements. The function is then called in the `__main__` block after taking three inputs from the user. The terminal window at the bottom shows the command `python c:/cygwin64/home/david/practica10/practica10.py` being executed, and the output `El numero es 6`.

```
File Edit Selection View Go Run Terminal Help
practica10.py X practica10while.py practica10bib.py practica10bib2.py p
C: > cygwin64 > home > david > practica10 > practica10.py > ...
1 def numeroMayor(a, b, c):
2     if a > b and a > c:
3         print("El numero es {}".format(a))
4     elif b > c and b > a:
5         print("El numero es {}".format(b))
6     else:
7         print("El numero es {}".format(c))
8
9 if __name__ == "__main__":
10     a = int(input())
11     b = int(input())
12     c = int(input())
13     numeroMayor(a, b, c)
14
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 3
PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10.py
4
5
6
El numero es 6
PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10.py
```

- PROGRAMA 2.

En este programa calculamos el factorial de cualquier número por medio de un while, tampoco hay mucha diferencia con este ciclo en C, solo que no se pudo realizar el do while y hablando un poco del programa en sí, no es posible aumentar el Contador con un incremento del tipo “i++”.



```
File Edit Selection View Go Run Terminal Help
practica10.py practica10while.py X practica10bib.py practica10bib2.py practica10bib3.py
C: > cygwin64 > home > david > practica10 > practica10while.py > ...
1 def factorial(n):
2     i = 2
3     temp =1
4     while i <= n:
5         temp = temp * i
6         i = i +1
7     return temp
8
9 if __name__ == "__main__":
10     a = int(input("Ingresa un numero : "))
11     print(factorial(a))

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 3
PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10while.py
Ingresa un numero : 4
24
PS C:\Users\david> 
```

- PROGRAMA 3.

En este programa realizamos los diferentes tipos de ciclos for que existen en Python, primero tenemos el for para listas simplemente declaramos nuestro for con ciertos argumentos y los va imprimiendo, el siguiente es el for para rangos, aquí es un poco diferente ya que le damos uno o dos argumentos, si le damos uno, nuestro for va a ir de 0 hasta el número que le dimos, si le damos dos números, este va a imprimir los números que estén dentro de ese rango y si le damos 3, este va a imprimir un rango desde el primer argumento, el segundo numero indicara cuantos espacios debe recorrer y el tercero de cuanto en cuanto los recorrerá.

El for para diccionarios funciona distinto ya que a cada elemento del diccionario se le asigna un valor numérico y por medio del for podemos imprimir el valor de cada uno por como si fuera una lista, también podemos imprimir el número directamente sin el element asignado y por ultimo también imprimimos el índice que ocupan los elementos o por decirlo de otra forma, el puesto que ocupa cada elemento dentro del diccionario, hay otros casos particulares como lo son el else for ya que aquí el for va a cumplir su ciclo hasta el rango que le demos, cuando este termine verá que no hay más casos, entonces pasará al else que es el caso que hay cuando el anterior termine.

```
File Edit Selection View Go Run Terminal Help
practica10.py practica10while.py practica10bib.py practica10t...
C: > cygwin64 > home > david > practica10 > practica10for.py > ...

1  #For para listas
2  def forlist():
3      for x in [1, 2, 3, 6, 5]:
4          print(x)
5
6  #For para rangos
7  def forrange():
8      for x in range(5):
9          print(x)
10
11     for y in range(-3, 3):
12         print(y)
13
14     for z in range(-4, 2, 2):
15         print(z)
16
17     for i in range(5, 0, -1):
18         print(i)
19
20 #For para diccionarios
21 def fordic():
22     diccionario = {'manzana':1, 'pera':3, 'uva':10}
23     for clave, valor in diccionario.items():
24         print(clave, " = ", valor)
25
26     for clave in diccionario.keys():
27         print(clave)
28
29     for valor in diccionario.values():
30         print(valor)
31
32     for idx, x in enumerate(diccionario):
33         print("El indice {} del elemento {}".format(idx, x))
34
35 #Else de for
36 def elsefor():
37     for x in range(5):
38         print(x)
39     else:
40         print("La cuenta se termino")
41
42
43 def elsefor2():
44     for x in range(5):
45         print(x)
46         if x== 2:
47             break
48     else:
49         print("la cuenta se termino")
50
51
52 #En else for 2
53 #Se rompio el ciclo y no entra al else
54
55 #calcular el factorial
56
57 if __name__ == "__main__":
```

Python 3.8.2 64-bit 3 0

```
File Edit Selection View Go Run Terminal Help
practica10.py practica10while.py practica10bib.py practica10bib2.py
C: > cygwin64 > home > david > practica10 > practica10for.py > forrange
4 print(x)
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 3
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10for.py
1
2
3
6
5
0
1
2
3
4
-3
-2
-1
0
1
2
-4
-2
0
5
4
3
2
1

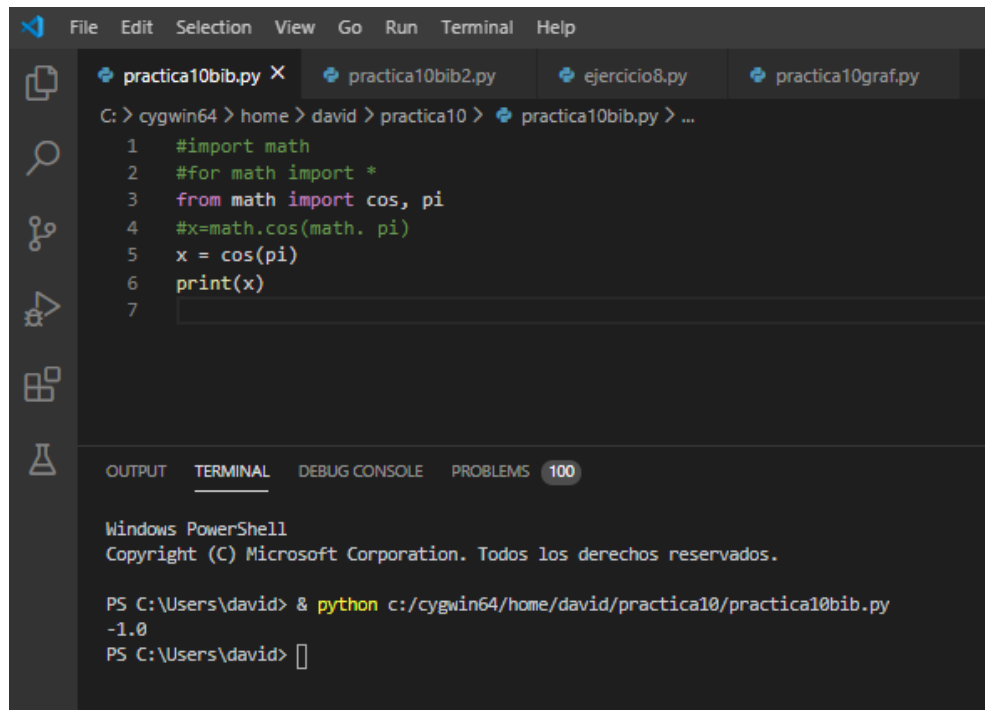
manzana = 1
pera = 3
uva = 10
manzana
pera
uva
1
3
10
El indice 0 del elemento manzana
El indice 1 del elemento pera
El indice 2 del elemento uva

0
1
2
3
4
La cuenta se termino

0
1
2
PS C:\Users\david> 
```

- PROGRAMA 4 y 5.

En estos dos programas simplemente probamos las maneras distintas de poder utilizar algunas librerías dentro de nuestros programas, las importábamos y las utilizamos de distintas maneras



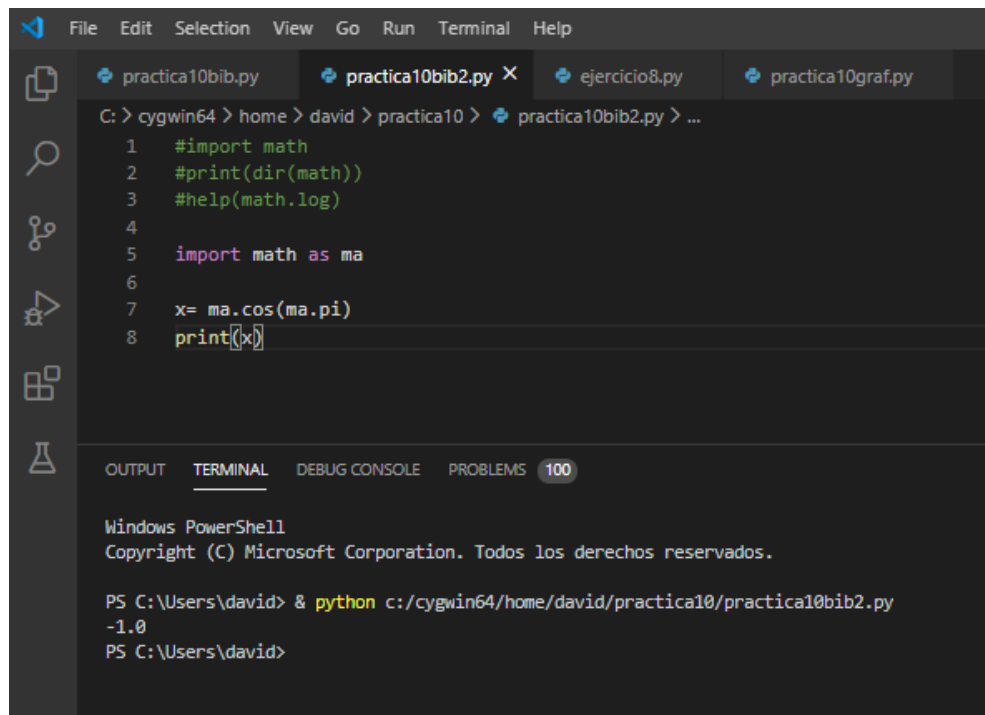
The screenshot shows the Visual Studio Code interface. The editor window displays the file `practica10bib.py` with the following Python code:

```
1 #import math
2 #for math import *
3 from math import cos, pi
4 #x=math.cos(math.pi)
5 x = cos(pi)
6 print(x)
7
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10bib.py
-1.0
PS C:\Users\david>
```



The screenshot shows the Visual Studio Code interface. The editor window displays the file `practica10bib2.py` with the following Python code:

```
1 #import math
2 #print(dir(math))
3 #help(math.log)
4
5 import math as ma
6
7 x= ma.cos(ma.pi)
8 print(x)
```

The bottom panel shows the `TERMINAL` tab with the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

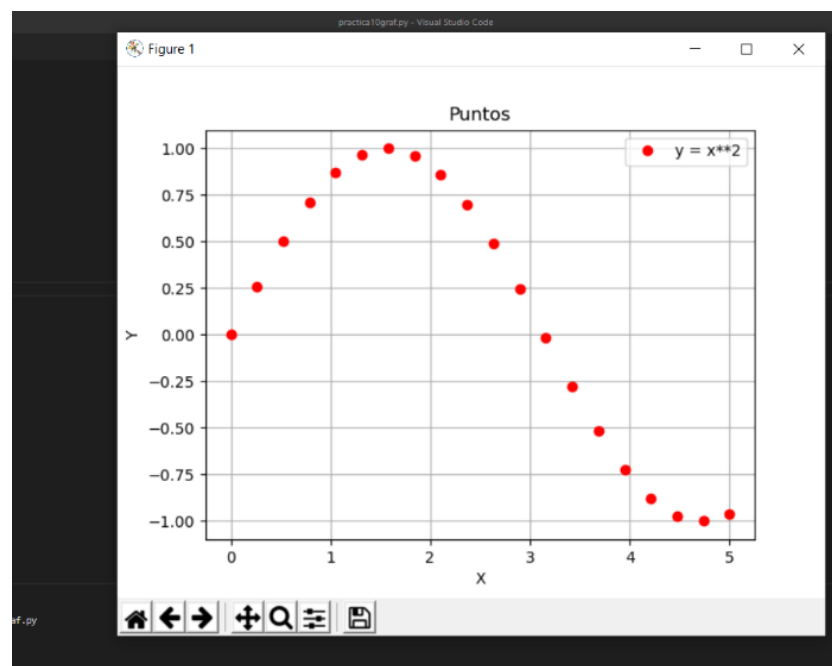
PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10bib2.py
-1.0
PS C:\Users\david>
```

PROGRAMA 6

En este programa graficamos por medio del uso de librerías la función seno de cero a 5 en el rango de las equis.

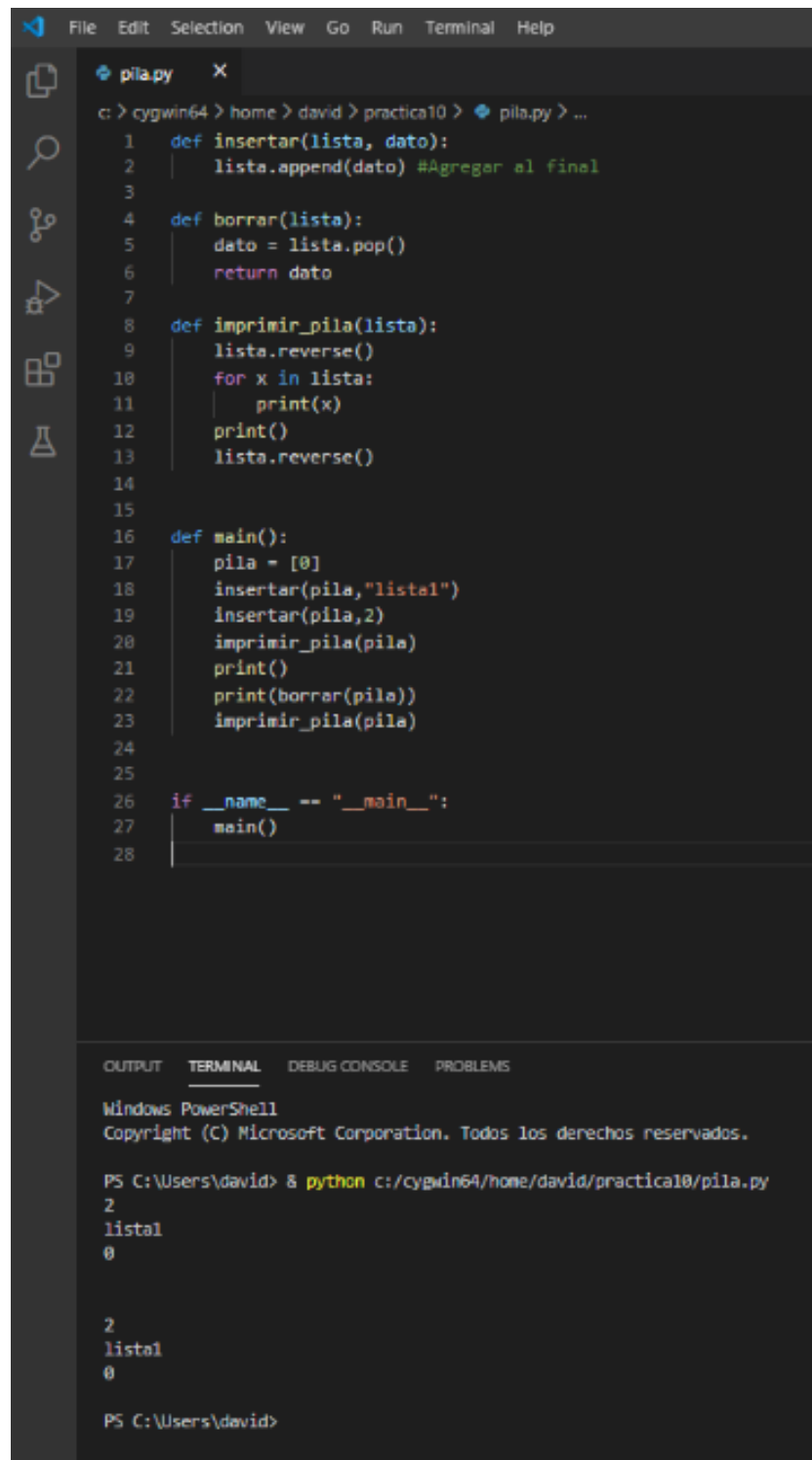
```
File Edit Selection View Go Run Terminal Help
practica10bib.py practica10bib2.py practica10graf.py X
C:\cygwin64\home\david\practica10> practica10graf.py > ...
1 import matplotlib.pyplot as plt
2 from mpl_toolkits.mplot3d import Axes3D
3 from numpy import *
4
5 x = linspace(0, 5, 20)
6
7 fig, ax = plt.subplots(facecolor='w', edgecolor='k')
8 ax.plot(x, sin(x), marker='o', color='r', linestyle='None')
9
10 ax.grid(True)
11 ax.set_xlabel('X') #Eje x
12 ax.set_ylabel('Y') #Eje y
13 ax.grid(True)
14 ax.legend(['y = x**2'])
15
16 plt.title('Puntos')
17 plt.show()
18
19 fig.savefig("grafica.png") #Guardando
20

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 100
PS C:\Users\david> & python c:/cygwin64/home/david/practica10/practica10graf.py
```



■ PROGRAMA 7

En este último programa hicimos una pila, la cual es mucho más sencilla de hacer que en C debido a que las funciones de una lista ya están creadas en Python, primero creamos una lista que agregara por el final a los datos y que fuera vaciando por el principio, lo que me pareció más impresionante de este programa es que realmente no se tuvieron que usar apuntadores y que le podíamos pasar cualquier tipo de dato a las listas.



```
File Edit Selection View Go Run Terminal Help

pila.py X
c: > cygwin64 > home > david > practica10 > pila.py > ...

1 def insertar(lista, dato):
2     lista.append(dato) #Agregar al final
3
4 def borrar(lista):
5     dato = lista.pop()
6     return dato
7
8 def imprimir_pila(lista):
9     lista.reverse()
10    for x in lista:
11        print(x)
12    print()
13    lista.reverse()
14
15
16 def main():
17     pila = []
18     insertar(pila, "lista1")
19     insertar(pila, 2)
20     imprimir_pila(pila)
21     print()
22     print(borrar(pila))
23     imprimir_pila(pila)
24
25
26 if __name__ == "__main__":
27     main()
28
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\david> & python c:/cygwin64/home/david/practica10/pila.py
2
lista1
0

2
lista1
0

PS C:\Users\david>

❖ CONCLUSIONES

-Esta práctica me pareció muy importante debido el uso que hicimos en Python, ya que no solamente vimos el uso de librerías, si no que los conceptos fueron bastante nuevos y además de eso pude hacer una comparación con C y notar las grandes diferencias, al igual de muchas facilidades con las que cuenta este lenguaje como lo es el uso de librerías para realizar gráficos y las funciones ya establecidas de una lista.