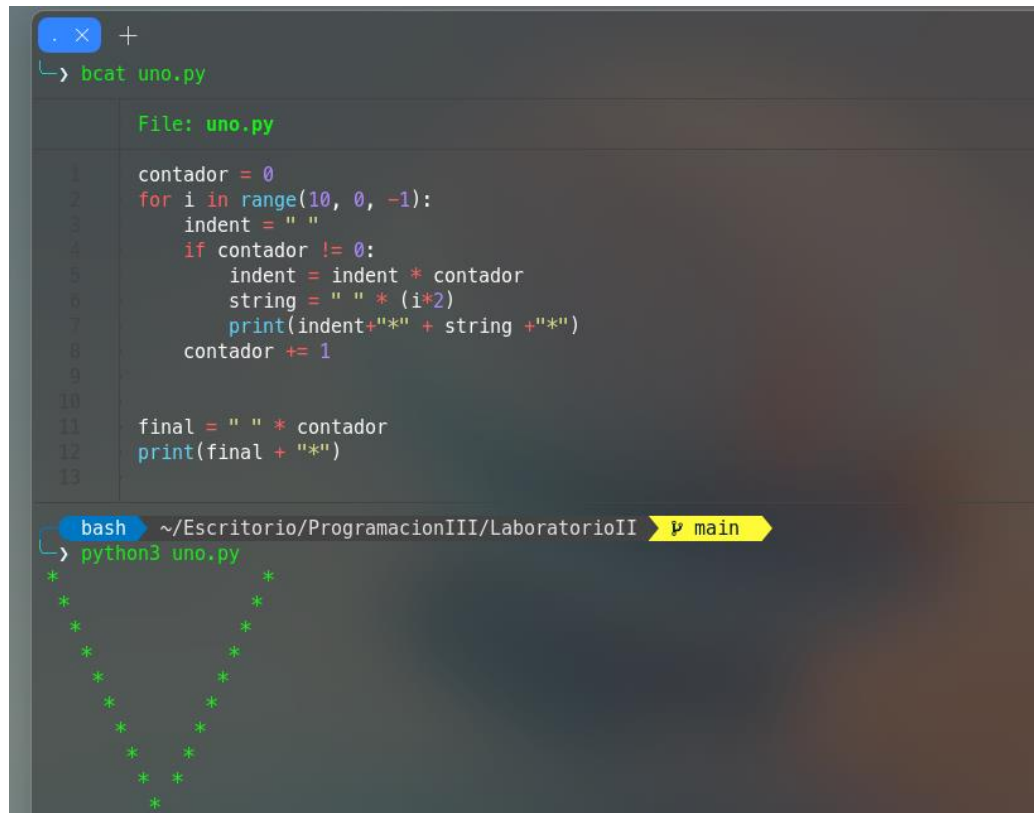


Universidad Mariano Gálvez
Facultad de Ingeniería
Ingeniería en Informática y Sistemas
Programación III

Laboratorio II

Oliver José Tzunun Dávila
0901-20-10107

- 1- La idea principal del primer ejercicio fue colocar una cadena de distancia entre los 2 primeros puntos, posteriormente ir escribiendo mas puntos y disminuyendo la cadena de distancia para llegar a colocar solo 1 punto, de esta manera formamos la V

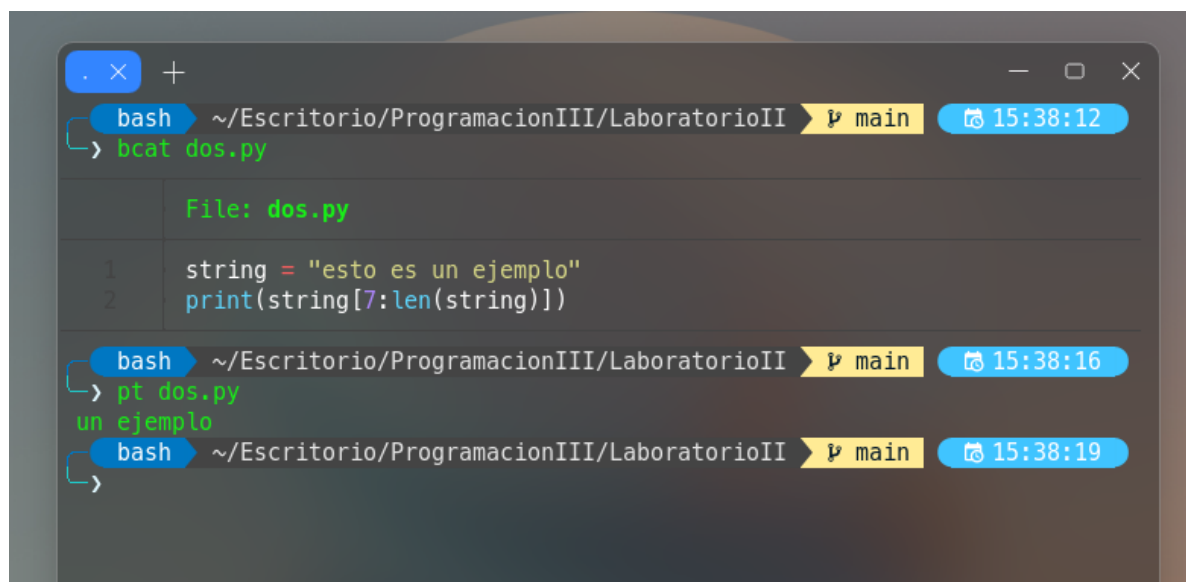


The screenshot shows a code editor with a file named `uno.py`. The code defines a function to print a 'V' shape using asterisks. It uses a loop to build a string of spaces and asterisks for each row, with the number of spaces decreasing and the number of asterisks increasing from top to bottom. The script is then executed in a terminal window.

```
File: uno.py
1 contador = 0
2 for i in range(10, 0, -1):
3     indent = " "
4     if contador != 0:
5         indent = indent * contador
6         string = " " * (i*2)
7         print(indent + "*" + string + "*")
8     contador += 1
9
10
11 final = " " * contador
12 print(final + "*")
13
```

```
bash ~/Escritorio/ProgramacionIII/LaboratorioII
> python3 uno.py
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
```

- 2- Lo que hicimos en este ejercicio, fue simplemente crear un substring o acortar la cadena principal, colocando como primer parámetro el punto "inicial: punto final"



The screenshot shows a code editor with a file named `dos.py`. The code creates a string and prints a substring starting from index 7 to the end. The script is then executed in a terminal window, showing the output 'un ejemplo'.

```
File: dos.py
1 string = "esto es un ejemplo"
2 print(string[7:len(string)])
```

```
bash ~/Escritorio/ProgramacionIII/LaboratorioII
> bcat dos.py

bash ~/Escritorio/ProgramacionIII/LaboratorioII
> pt dos.py
un ejemplo

bash ~/Escritorio/ProgramacionIII/LaboratorioII
>
```

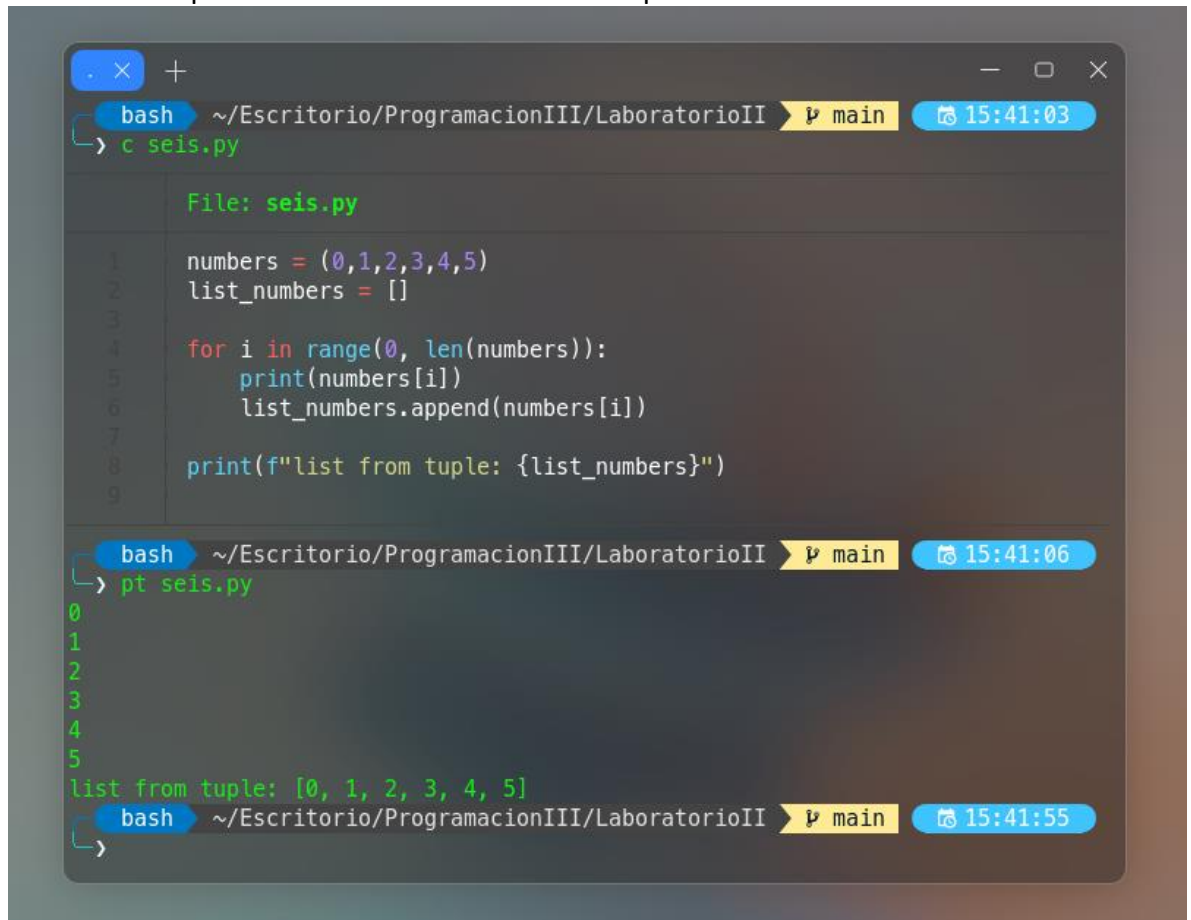
- 3- Para este ejercicio simplemente determinamos en que posición de la lista padre se encontraba el arreglo a modificar, una vez teniendo en cuenta la posición "1" accedimos a esa posición e hicimos la inserción del nuevo dato, dato del tipo lista también

```
bash ~/Escritorio/ProgramacionIII/LaboratorioII > c tres.py
File: tres.py
1 a=[1,[2,[3,4]],5]
2
3 a[1].append([6,7])
4
5 print(a)
bash ~/Escritorio/ProgramacionIII/LaboratorioII > pt tres.py
[1, [2, [3, 4], [6, 7]], 5]
bash ~/Escritorio/ProgramacionIII/LaboratorioII >
```

- 4- Utilizamos algo llamado slicing que consiste en extraer elementos de una secuencia, en este caso una secuencia de caracteres que conforman una cadena o un "string" de esta manera extraemos solo una pequeña parte de este grupo, diciéndole de donde queremos comenzar y donde terminal la parte a extraer
- 5- Colocamos el rango en el cual trabajaremos (1, 10) en cada iteración agregamos el elemento a la lista "numbers" y de ultimo imprimimos todo el contenido de la lista

```
bash ~/Escritorio/ProgramacionIII/LaboratorioII > c cinco.py
File: cinco.py
1 numbers = []
2 for i in range(1, 11): numbers.append(i)
3 print("Numbers in list \n")
4 print(numbers)
bash ~/Escritorio/ProgramacionIII/LaboratorioII > pt cinco.py
Numbers in list
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
bash ~/Escritorio/ProgramacionIII/LaboratorioII >
```

- 6- Tenemos una tupla de 6 elementos, creamos una lista vacia a la cual insertaremos todos los elementos de la tupla, esta vez decidimos acceder a los elementos de la tupla por medio de su índice, por lo que creamos un bucle cuya variable de manejo será la "i" que ira de (0 a size) donde "size" es el tamaño de la tupla, posteriormente imprimimos cada elemento de la tupla y luego lo ingresamos a la lista y asi finalmente imprimir la lista con los datos de la tupla



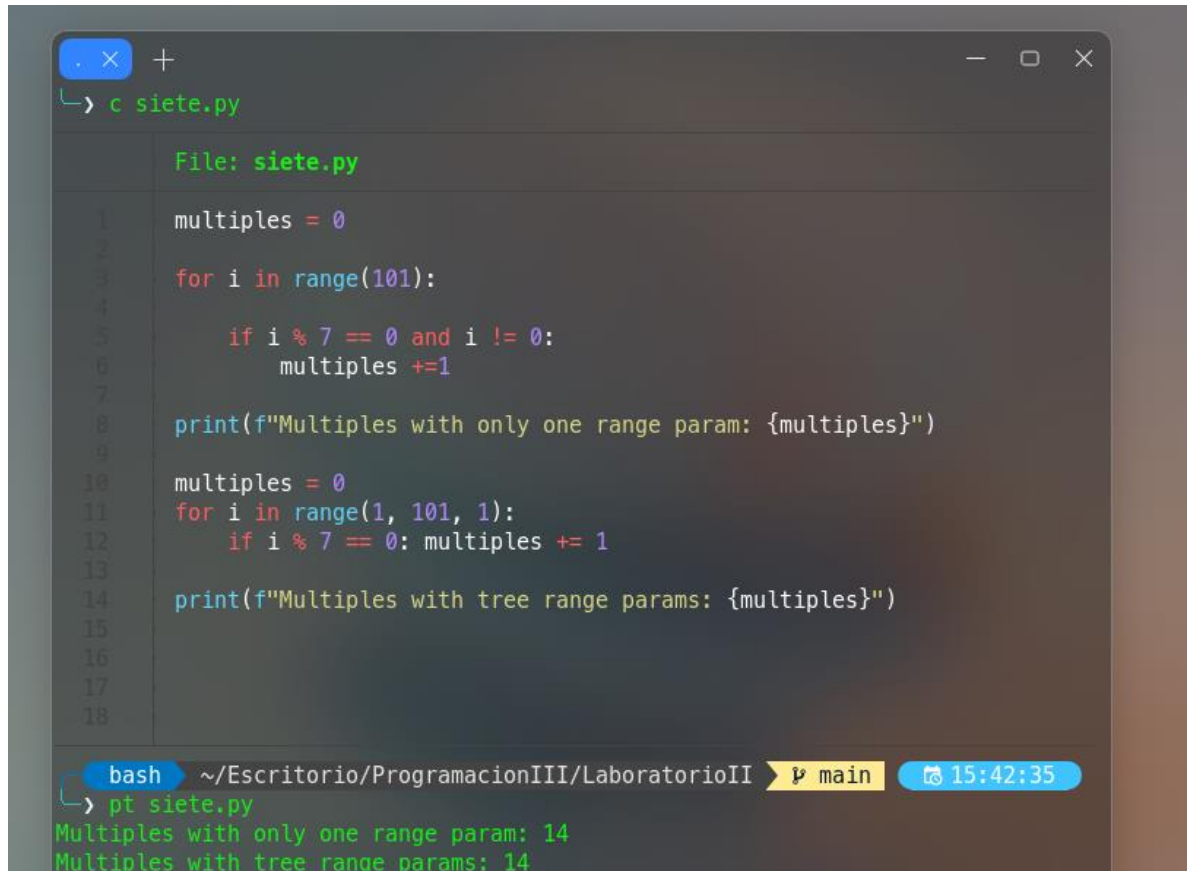
```
bash ~/Escritorio/ProgramacionIII/LaboratorioII main 15:41:03
> c seis.py

File: seis.py
1 numbers = (0,1,2,3,4,5)
2 list_numbers = []
3
4 for i in range(0, len(numbers)):
5     print(numbers[i])
6     list_numbers.append(numbers[i])
7
8 print(f"list from tuple: {list_numbers}")
9

bash ~/Escritorio/ProgramacionIII/LaboratorioII main 15:41:06
> pt seis.py
0
1
2
3
4
5
list from tuple: [0, 1, 2, 3, 4, 5]

bash ~/Escritorio/ProgramacionIII/LaboratorioII main 15:41:55
>
```

- 7- Creamos un contador de los múltiplos, comenzamos a iterar desde 0 hasta 100, verificamos que no estemos en la posición 0 debido a que esto nos da 0 como residuo, luego si el numero que estamos iterando es multiplo de 7 incrementamos los múltiplos, al final imprimimos y hacemos lo mismo en el segundo for, reseteando en 0 nuestro contador de múltiplos y luego realizamos lo mismo, solo que le decimos que comenzaremos en 1 y terminaremos en 100, para ir incrementando de 1 en 1



```
File: siete.py

1 multiples = 0
2
3 for i in range(101):
4
5     if i % 7 == 0 and i != 0:
6         multiples +=1
7
8 print(f"Multiples with only one range param: {multiples}")
9
10 multiples = 0
11 for i in range(1, 101, 1):
12     if i % 7 == 0: multiples += 1
13
14 print(f"Multiples with tree range params: {multiples}")
15
16
17
18

bash ~/Escritorio/ProgramacionIII/LaboratorioII main 15:42:35
> pt siete.py
Multiples with only one range param: 14
Multiples with tree range params: 14
```

Repositorio de GitHub: <https://github.com/Olivers11/LaboratorioII>