

- 1) Meter los números del 1 al 20 en una lista y mostrarla en pantalla. Hacer lo mismo para un rango de números indicado por un usuario.
- 2) Pide un número y guarda en una lista su tabla de multiplicar hasta el 10. Por ejemplo, si pide el 5 la lista tendrá: 5,10,15,20,25,30,35,40,45,50
- 3) Pide una cadena (string) por teclado, mete los caracteres en una lista sin repetir caracteres.
- 4) Pide una cadena (string) por teclado, mete los caracteres en una lista sin espacios.
  
- 5) Crea una tupla con números, pide un numero por teclado e indica cuantas veces se repite.
- 6) Crea una tupla con los meses del año, pedir números al usuario. Si el numero esta entre 1 y la longitud máxima de la tupla, muestra el contenido de esa posición sino muestra un mensaje de error. El programa termina cuando el usuario introduce un cero
- 7) Crea una tupla con números e indica el número con mayor valor y el que menor tenga.
  
- 8) Escribir una función que reciba una cadena y devuelva un diccionario con la cantidad de apariciones de cada palabra en la cadena. Por ejemplo, si recibe "Qué lindo día que hace hoy" debe devolver: 'que': 2, 'lindo': 1, 'día': 1, 'hace': 1, 'hoy': 1
- 9) Escribir un programa que vaya solicitando al usuario que ingrese nombres.
  - Si el nombre se encuentra en la agenda (implementada con un diccionario), debe mostrar el teléfono y, opcionalmente, permitir modificarlo si no es correcto.
  - Si el nombre no se encuentra, debe permitir ingresar el teléfono correspondiente. El usuario puede utilizar la cadena "\*", para salir del programa
  
- 10) Opcional: Pide números y mételos en una lista, cuando el usuario meta un 0 ya dejaremos de insertar. Por último, muestra los números ordenados de menor a mayor.
- 11) Opcional: Lo mismo que el anterior, pero ordenando de mayor a menor.
  
- 12) Opcional: Ta-te-ti  
Desarrolle un programa en Python que permita a dos personas jugar al ta-te-ti. Haga que el jugador 1 sea X y que el jugador 2 sea O. Haga que en su turno cada jugador indique donde quiere colocar su símbolo. El programa debería terminar si alguien gana o hay un empate. También debe mostrar el tablero en pantalla luego de cada movimiento.

X	--	X
O	O	--
X	X	O
  
- 13) Opcional: Codificador Morse: Desarrolle un programa en Python que permita al usuario escribir un mensaje y convertirlo a código Morse. La codificación Morse se presenta en la siguiente tabla:

<b>A</b>	<b>· -</b>	<b>N</b>	<b>- ·</b>
<b>B</b>	<b>- · · ·</b>	<b>O</b>	<b>- - -</b>
<b>C</b>	<b>- · - ·</b>	<b>P</b>	<b>· - - -</b>
<b>D</b>	<b>- · ·</b>	<b>Q</b>	<b>- - - -</b>
<b>E</b>	<b>·</b>	<b>R</b>	<b>· - ·</b>
<b>F</b>	<b>· · - ·</b>	<b>S</b>	<b>· · ·</b>
<b>G</b>	<b>- - ·</b>	<b>T</b>	<b>-</b>
<b>H</b>	<b>· · · ·</b>	<b>U</b>	<b>· · -</b>
<b>I</b>	<b>· ·</b>	<b>V</b>	<b>· · · -</b>
<b>J</b>	<b>· - - -</b>	<b>W</b>	<b>· - -</b>
<b>K</b>	<b>- · -</b>	<b>X</b>	<b>- · · -</b>
<b>L</b>	<b>· - · ·</b>	<b>Y</b>	<b>- · - -</b>
<b>M</b>	<b>- -</b>	<b>Z</b>	<b>- - · ·</b>

Muestre el mensaje codificado de manera tal que haya una letra en Morse por línea, y separe las palabras con dos líneas en blanco. Por ejemplo, ‘hola mundo’ se mostraría:

```

....
---
·-·
·-

--
·-
-·
-·
---
```