

Práctico II

Introducción a secuencias ordenadas (*Strings, Listas*)

Los ejercicios a continuación utilizarán variables, posiblemente estructuras de decisión y estructura de repetición for. Pero no todos, ni siempre. Debe determinar en cada caso, qué estructura será necesaria, si harán falta combinaciones de estructuras o no, etc.

1. Guarde en una variable, la palabra 'Mordor'. Muestre en pantalla las letras en las posiciones 1 y 3. ¿Qué letras resultan ser?
2. Pida una frase al usuario. Muestre su primer letra en pantalla. Muestrela en mayúsculas también.
3. Pida una frase al usuario, controle que tenga una longitud total mayor a 5 caracteres. Muestre en pantalla los primeros 3 caracteres de la misma.
4. Pida un número al usuario, mayor que 1 y menor a 50. Muestre en pantalla los números de 1 hasta ese número ingresado, uno por línea.
5. Pedir dos palabras al usuario y mostrarlas en pantalla concatenadas, es decir, una seguida de la otra. ¿Cuáles son todas las maneras en que se pueden mostrar concatenadas en pantalla, cadenas de caracteres? ¿Qué diferencia hay entre mostrarlas una seguida de otra en pantalla, y en concatenarlas?

Más avanzados

Cadenas de Caracteres o Strings

6. El usuario podrá ingresar nombre y apellido. El programa deberá convertir las iniciales en mayúsculas y las demás letras en minúsculas. Mostrar por pantalla.
7. Se le permitirá al usuario ingresar una frase. Se mostrarán en pantalla solamente las letras en posiciones pares de la misma.
8. El usuario deberá ingresar una frase y una letra. Determine cuántas veces está esa letra en dicha frase, mostrar por pantalla.
9. Permita al usuario ingresar una frase. Cambie las letras *a* por *4* y las letras *e* por *3*.
10. Permita al usuario ingresar el nombre de un archivo, genere un nuevo nombre donde los espacios sean reemplazados por guión bajo y la extensión por numerales.
11. Permitir ingresar al usuario un número de un dígito. Controlando se haya ingresado dicho número de no más de 1 dígito de longitud, pasarlo a letras y mostrarlo en pantalla. (*Ejemplo: Si ingresa 3, se verá como resultado "tres"*).

-
12. Se le pedirá al usuario una frase. Se mostrarán en pantalla, una palabra por línea de la misma. **no usar listas en este ejercicio*

Listas

13. Dada la siguiente lista de valores numéricos [2, 65, 34, 3, 8, 65] Realice la suma de los elementos que están ubicados en las posiciones 0, 2 y 5 Muestre el resultado en pantalla.
14. Dada la siguiente lista de compras de ingredientes para preparar una torta, mostrarla en pantalla, un ingrediente por línea. Luego corregir el último a "Canela en polvo" ["Chocolate", "Huevos", "Manteca", "Crema de leche", "Frutillas"]
15. Dada la siguiente lista de valores numéricos [56, 7, 34, 19, 3, 1, 76, 2, 81, 4, 2, 8] muestre en pantalla solo los elementos de la misma que están ubicados en posiciones pares, como 0, 2, 4, etc (*¿Cómo puede determinar si un nro es par o no? ¿Deberá escribir cada print de a uno, o deberá considerar realizar un recorrido por la lista, usando un bucle?*)
16. Guarde en dos variables de listas diferentes los siguientes elementos: ["amarillo", "azul", "violeta"] y ["zapallo", "tomate", "limón"] Genere una nueva lista que sea la concatenación de ambas, le deberá quedar ["amarillo", "azul", "violeta", "zapallo", "tomate", "limón"]
17. Pida 10 nombres de películas al usuario. Guardelos en una lista. Luego pida al usuario que ingrese un número *n* del 1 al 10. Controle que *n* esté en el rango correcto, es decir entre 1 y 10. Muestre en pantalla cuál es la película *n*-ésima. Por ejemplo, si el usuario me ingresa 1, debo mostrar la primer película de la lista.
18. Crear dos listas del mismo tamaño, y luego armar una tercer lista, a la cual primero se le agregue el primer elemento de la lista 1, luego el primer elemento de la lista 2. Luego se le agregue el segundo elemento de la lista 1, luego el segundo elemento de la lista 2, y así sucesivamente.
Ejemplo: Supongamos tenemos una lista de frutas y otra de verduras. La tercera quedará:
- ```
[‘damasco’, ‘frutilla’, ‘ananá’] [‘zamahoria’,
‘berenjena’, ‘tomate’]

[‘damasco’, ‘zamahoria’, ‘frutilla’, ‘berenjena’, ‘ananá’, ‘tomate’]
```
19. Pedir el nombre al usuario, y corroborar si ese nombre existe entre los nombres de usuarios válidos guardados en una lista.
20. Realizar un programa que le pida al usuario ingresar una frase al usuario y coloque cada palabra de la misma como elemento de una lista.
21. Realizar un programa que le pida ingresar una frase al usuario y coloque cada letra como elemento de una lista.

- 
22. El usuario deberá poder ingresar varios nombres completos (*ejemplo: "Luis Pérez"*). El programa deberá luego, colocar los nombres en una lista y los apellidos en otra.
  23. Se deberán ingresar 8 notas a una lista. Se mostrará el promedio, redondeado a 2 decimales.
  24. Pedir al usuario una frase. Determinar de al menos dos modos diferentes (*con y sin listas*) la cantidad de palabras que hay en dicha frase.
  25. Dada una lista de números, ingresada por el usuario o inventada por usted, cree otra lista con la cantidad de dígitos de cada número de la misma.

### Pseudo Aleatoriedad

26. Tirar 20 veces un dado de 6 caras. Mostrar el promedio de esas 20 tiradas.
27. Tirar ahora, 2500 veces un dado de 6 caras. Mostrar el promedio de esas tiradas. Comparar con el promedio del ejercicio anterior. ¿Nota una diferencia sustancial habiendo cambiado la cantidad de tiradas?
28. Pedirle al usuario sus 10 marcas favoritas. Mostrar una marca al azar de la lista.

### Variado

29. Pedirle al usuario la cantidad de notas que desea ingresar. Luego pedir cada nota, y guardarlas.
30. Cuenta Regresiva: Se requiere un programa que permita el ingreso de un número positivo y muestre en pantalla la cuenta regresiva desde el número ingresado hasta llegar a 0. Realizar diferentes versiones del programa, utilizando en cada una, una estructura de bucle diferente de las que tiene disponibles en Python.
31. Suma de Números Positivos y Negativos: Se requiere un programa que permita el ingreso de 10 números y al finalizar muestre en pantalla la cantidad números positivos y por otra parte la cantidad de números negativos que fueron ingresados.
32. Permitir ingresar números enteros hasta que se ingrese la opción "s" de salir. Luego mostrar por pantalla los números ingresados.
33. Desarrollar un programa que permita al usuario ingresar una serie de números, sumar todos los pares y al terminar mostrar dicha suma. Si se ingresó algún impar, mostrar un mensaje "*Se ingresaron impares*". Para finalizar el ingreso, indicar la cantidad de números a ingresar al principio del programa, o interrumpir la carga cuando se ingrese el número 99.
34. Permitir ingresar 10 números al usuario. Determinar y mostrar el menor y el mayor.
35. Número Invertido: Se requiere mostrar en pantalla un número invertido de 6 cifras, al que fuera ingresado por teclado. (*Ejemplo: en pantalla se verá: "El número ingresado es 140975, invertido es: 579041"*)

- 
36. Pedirle al usuario dos números positivos,  $a$  y  $b$ . Controlar que  $a < b$ . Mostrar en pantalla los números del intervalo cerrado  $[a, b]$ . La computadora deberá ahora seleccionar al azar un número de ese intervalo. Y el usuario deberá adivinar cuál número ha sido seleccionado por la computadora, obteniendo un mensaje de éxito en caso de acertar. El usuario solo tendrá 10 vidas (*número de intentos*) y en caso de no acertar, deberá obtener un mensaje de pucha.

