

Guía 6 - Ciclos Definidos

(★, ★★, ★★★) Esta notación indica la dificultad de cada ejercicio, de menor a mayor.

El documento FE DE ERRATA contiene todos los errores que han sido detectados en las Resoluciones de las Guías de Ejercicios y sus correcciones. Si detectan un error nuevo en una Resolución, por favor enviar un mail a tcasiraghi@udesa.edu.ar indicando el error.

Temas incluidos en esta guía

- Ciclos Definidos

Ejercicios

1. ★ Escriba un programa que recorra la siguiente lista usando el comando for e imprima un mensaje si el número es o no es par. Debe decir algo por el estilo de:

El número [NUMERO] es par

o

El número [NUMERO] es impar

```
1 numeros = [934, 845, 936, 791, 887, 233, 820, 366, 587, 104]
2
```

2. ★ Escriba un programa que recorra la siguiente lista e imprima en pantalla el cuadrado de cada número. Si en algún momento se encuentra un número negativo el programa debe:

- Versión 1: Ignorarlo y continuar con los otros números
- Versión 2: Imprimir “!Se encontró un número negativo!” y no continuar con la lista de números

```
1 lista = [2, 5, 3, 9, 12, 1, -5, 7, 10, -8, 12]
2
```

3. ★ Escriba un programa que recorra un texto ingresado por el usuario y que genere un texto igual pero con todas las “s” en mayúscula.

Nota: No use el método replace

4. ★ Escriba un programa que recorra el diccionario e imprima las claves que guardan numeros pares.

```
1 my_diccio = {'Lunes':3, 'Martes':6, 'Miércoles':204232,
2             'Jueves':49231, 'Viernes':23218, 'Sábado':321,
3             'Domingo':6}
4
```

5. ★ Haga un programa que imprima los nombres que comienzan con la letra "a" ó "A" en la siguiente lista.

```
1 nombres = ["Ambrosia", "Humberto", "Antonio", "Diógenes"]
2
```

6. ★ Escribir un programa que imprima los nombres de longitud menor a 6. Por ejemplo, dada la siguiente lista:

```

1 nombres = ["John", "Polly", "Thomas", "Arthur", "Michael", "Finn", "Ada"]
2

```

Se debería imprimir

```

1 John
2 Polly
3 Finn
4 Ada
5

```

7. ★ Escribir un programa que cuente la cantidad de nombres de longitud menor a 6 e imprima el resultado. Por ejemplo, dada la siguiente lista:

```

1 nombres = ["John", "Polly", "Thomas", "Arthur", "Michael", "Finn", "Ada"]
2

```

Se debería imprimir 4

8. ★ ★ Escribir un programa que imprima una **lista nueva** con nombres de longitud menor a 6. Por ejemplo, dada la siguiente lista:

```

1 nombres = ["John", "Polly", "Thomas", "Arthur", "Michael", "Finn", "Ada"]
2

```

Se deberían imprimir ['John', 'Polly', 'Finn', 'Ada']

9. ★ Escriba un programa que recorra solo los elementos en índices impares de la siguiente tupla y que guarde en otra lista el triple de cada número. Por ejemplo si la tupla fuera (1,2,3,4,5,6) debería generar una lista como [6,12,18]

```

1 numeros = (934, 845, 936, 791, 887, 233, 820, 366, 587, 104)
2

```

10. ★ ★ Escriba un programa que dada una lista de números arbitraria y un número solicitado al usuario, agregue el doble de ese número en la posición anterior a la que aparece en la lista. Si el número no está en la lista no se debería modificar nada. Por ejemplo, si el usuario ingresara el número 6 y la lista fuera:

```

1 l = [1,3,5,6,7,6]
2

```

Se debería imprimir:

```

1 [1,3,5,12,6,7,12,6]
2

```

11. ★ ¿Cuántas veces se imprime en pantalla con este código? ¿Porqué?

```

1 lista = [1, 2, 3]
2 for i in lista:
3     for j in lista:
4         for k in lista:
5             print(i,j,k)
6

```

12. ★ ★ Escriba un programa que recorra un texto ingresado por el usuario y que genere un texto igual pero con todas las vocales en mayúscula.

Nota: cuidado con las vocales con tilde. Tal vez les sea útil el comando “in”

13. ★ ¿Qué imprime el siguiente programa?

```
1 for n in range(5):
2     print(n)
3
```

14. ★ Reemplazar X e Y en el siguiente programa para que se impriman los números del 4 al 10, incluyendo al 10.

```
1 for n in range(X, Y):
2     print(n)
3
```

15. ★ Reemplazar X, Y, Z en el siguiente programa para que se impriman los números del 4 al 10, incluyendo al 10, de 2 en 2. Es decir: 4,6,8,10

```
1 for n in range(X, Y, Z):
2     print(n)
3
```

16. ★ `arboles_por_km2` es un diccionario que contiene como claves los nombres de distintos países y como valores asociados, la cantidad de árboles por kilómetro cuadrado que posee dicho país:

```
1 arboles_por_km2 = {"Brasil": 39542, "Finlandia": 90652,
2                   "Estados Unidos": 23513, "Dinamarca": 6129,
3                   "Siria": 534, "Arabia Saudita": 1}
4
```

Escriba un programa que evalúe cada clave del diccionario e imprima ‘[PAÍS] tiene más de 20000 árboles por kilómetro cuadrado’ si el país en cuestión tiene más de 20000 árboles por metro cuadrado, y ‘[PAÍS] tiene menos de 20000 árboles por kilómetro cuadrado’ en el caso contrario.

Por ejemplo, al evaluar la clave ‘Finlandia’ el programa debería imprimir ‘Finlandia tiene más de 20000 árboles por kilómetro cuadrado’, mientras que en el caso de ‘Siria’ se debería mostrar ‘Siria tiene menos de 20000 árboles por kilómetro cuadrado’.

Nota: Use un for con los items del diccionario!

17. ★ ¿Los siguientes 3 programas hacen lo mismo? ¿Por qué?

```
1 for n in range(1, 10):
2     print(n)
3
```

```
1 for numero in range(1, 10):
2     print(numero)
3
```

```
1 for avion in range(1, 10):
2     print(avion)
3
```

18. ★ Escribir un programa que imprima una lista con los números pares del 1 al 15, una vez usando el comando for, y otra vez sin usar el comando for.

19. ★ Escribir un programa que imprima (uno por línea) el nombre del alumno y su nota correspondiente. Estos datos están guardados en dos listas separadas como se ve. El orden de notas es igual al orden de los alumnos. Haga una versión con y una sin la función zip()

```
1 alumnos = ['Juan', 'Roberto', 'Ignacio', 'Claudio', 'Patricio']
2 notas = [10, 6, 3, 8, 4]
3
```

20. ★ Escribir una programa que imprima una lista con las posiciones de los números menores a 4 de la siguiente lista.

Hint: Utilice la función enumerate().

```
1 notas = [6, 0, 10, 8, 4, 6, 9, 8, 2, 0, 9, 7, 6, 4, 0, 1, 5, 0, 0, 1]
2
```

21. ★ ★ Se tienen tres listas. La lista “nombres” guarda los nombres de cada jugador del torneo. La lista “total_puntajes” guarda el total de puntaje obtenido por cada jugador durante todo el torneo. La lista “partidos_jugados” guarda la cantidad de partidos jugados por cada jugador. Imprima en pantalla el nombre del jugador que obtuvo en el puntaje promedio mas alto (puntaje promedio = puntaje total / partidos jugados).

Tip: le puede ser útil la función zip()

```
1 nombres = ['Juan', 'Roberto', 'Ignacio', 'Claudio', 'Patricio']
2 puntaje_total = [107, 632, 234, 180, 61]
3 partidos_jugados = [4, 25, 8, 7, 2]
4
```

22. ★ ★ ★ Se le pide a un programador que haga un código para buscar numeros pares en una lista y eliminarlos. El programador hace el siguiente código pero por alguna razon no parece funcionarle para ciertas listas. Por ejemplo, si la lista es [3, 7, 4, 9, 7, 2, 10, 6, 6, 7] se espera que devuelva [3, 7, 9, 7, 7]. Sin embargo el código devuelve [3, 7, 9, 7, 10, 6, 7]. Si bien se eliminó el 4, el 2 y uno de los 6 todavia quedan el 10 y el otro 6. Encuentre el error en el código y proponga una solución.

```
1 lista_de_numeros = [3, 7, 4, 9, 7, 2, 10, 6, 6, 7]
2 for index, numero in enumerate(lista_de_numeros):
3     if numero%2 == 0:
4         lista_de_numeros.pop(index)
5 print(lista_de_numeros)
6
```

23. ★ ★ ★ Escriba un programa que implemente un traductor, para esto cuenta con un diccionario que tiene como claves las palabras escritas en inglés, y como valores su correspondiente traducción al español. El programa le debe solicitar al usuario que ingrese una frase escrita en inglés, y debe imprimir la frase traducida al español (para probar solo utilice pocas palabras en el diccionario). Si la palabra en inglés no está en el diccionario, se debe dejar escrita en inglés en el resultado final.

Nota: Asuma que todas las palabras estan separadas por espacio y que no hay signos de puntuación

24. ★ ★ ★ Escriba un programa que le pida al usuario un texto. Su código debe recorrer este texto, remover todos los signos de puntuación y luego imprimir el texto “limpio”

Hint: Les puede servir utilizar el método `isalnum()`

25. ★ ★ ★ Se tienen tres listas de datos, una denominada “DNI” que contiene los DNI de 20 personas, la otra, denominada “Nombres”, contiene sus nombres, la tercera, “Apellidos”, contiene sus apellidos. Se necesita crear un diccionario que como clave tenga los DNI de cada persona, y como valor otro diccionario. Este segundo diccionario tiene las claves “nombre” y “apellido” que guardan el nombre y apellido.

```
1 DNI = [33694911, 44328642, 31643135, 35241269, 45542153, 17037523,
2       32238035, 21355847, 11496916, 28619955, 29226865, 39449209,
3       22703107, 23184773, 27041529, 17973562, 27451433, 29996758,
4       44633758, 38731825]
5
6 Nombres = [Fabiana, Lucas, Luciano, Isidora, Agustín, Antonio, Horacio,
7            Benjamín, Bruno, Anthony, Isaac, Emilio, Leonardo, Carla,
8            Alejandra, Isabel, Valery, Juan Pablo, Gael, Alexa]
9
10 Apellidos = [Garrido, Granados, Griego, Leon, Guerrero, Rojas, Nieto,
11              Guerrero, Herrera, Flores, Vicente, Carrasco, Dominguez,
12              Torres, Moya, Alvarez, Enríquez, Martín, Díaz, Santiago]
13
```

26. ★ ★ ★ Escribir un programa que dada una lista de strings que son palabras a buscar (`palabras_a_buscar`) y un string largo (una frase sin signos de puntuación), genere un diccionario donde las claves son cada una de las palabras a buscar y el valor es la cantidad de veces que la palabra aparece en el string largo (frase). Por ejemplo, dada la lista:

```
1 palabras_a_buscar = ["yo", "manejo", "rolls-royce", "perro"]
2
```

y la frase

```
1 frase = "yo no manejo el rating yo manejo un rolls-royce"
2
```

El programa debe imprimir el diccionario

```
1 {"yo": 2, "manejo": 2, "rolls-royce": 1, "perro": 0}
2
```

Hint: Les podría servir el método `split()`

27. ★ ★ ★ Escribir un programa que dados un diccionario con el listado de un carrito de supermercado y otro con los precios de los productos de un supermercado, imprima el costo total del carrito. **Aclaración:** el programa debe poder correr para cualquier carrito de supermercado. Es decir, que si un producto del carrito no existe dentro del listado de precios, el programa debe imprimir el mensaje de error “El producto X no existe en el supermercado” y este no debe ser incluido al costo total del carrito. Por ejemplo, dado el carrito:

```
1 carrito = {
2     "topline 7": 2,
3     "fanta": 1,
4     "lata de atun": 6,
5     "esparragos": 100,
6 }
7
```

y los precios

```
1  precios = {  
2      "topline 7": 50,  
3      "beldent": 25,  
4      "fanta": 200,  
5      "coca light": 140,  
6      "coca comun": 140,  
7      "lata de atun": 150,  
8  }  
9
```

El programa debe imprimir:

El producto esparragos no existe en el supermercado
El costo total del carrito es: \$1200

La cuenta que se realiza es:
2 latas de atún * \$50 +
1 botella de fanta * \$200 +
6 latas de atún * \$150
= \$1200