

UNIVERSIDAD ESCUELA COLOMBIANA DE INGENIERÍA

Asignatura: Algoritmos y Programación (AYPR)

Profesora: Ingeniera Patricia Salazar Perdomo

Fecha: 19 de noviembre de 2024

Ref.: Ejercicios No. 8. Programación modular con listas.

Cada archivo se debe llamar EJ8.# Apellidos Nombre, Apellidos Nombre, donde # estará entre 1 y 5, según corresponda. Si trabaja solo, entonces EJ8.# Apellidos Nombre
El archivo comprimido se debe llamar EJ8 Apellidos Nombre, Apellidos Nombre o EJ8.# Apellidos Nombre, según corresponda.
Les doy los esqueletos.

- Construir **una función** en Python que solucione cada uno de los problemas que se dan a continuación.
- Escribir un ejemplo de invocación de cada función. Hay que darles nombres significativos a las variables y a las funciones.
- Hacer pruebas suficientes a cada función para asegurarse de que hacen lo que se pide.
- **No se pide la solución modular completa.**

Importante:

El tema es listas. Por tanto, debe demostrar dominio del tema cubierto hasta el momento, entre otros, el recorrido por contenido.

1. **Suponga que se tiene una lista** con los nombres y los precios de varios útiles escolares.

Por ejemplo:

```
utiles_esc = [["lápiz", 1500], ["cuaderno", 12200], ["esfero", 3700], ["borrador", 1000], ["regla", 5400], ["tajalápiz", 2700], ["block de papel periódico", 7500], ["colores", 42000], ["corrector", 6800], ["cartulina", 2100], ["escuadra", 4800]]
```

Construya **una función** en Python que inicialice dos listas: una con los útiles y otra con los precios.

Las listas deben quedar disponibles en la función que llame a ésta. Los datos pueden ser los del ejemplo u otros con la misma estructura.

2. Los tipos de sangre son O, A, B y AB. El factor Rhesus (**Rh**) es una proteína heredada que se encuentra en la superficie de los glóbulos rojos. Si la sangre contiene esta proteína, la persona es **Rh** positivo. Si la sangre carece de ella, es **Rh** negativo. **Rh** positivo es el grupo sanguíneo más frecuente. Tomado de https://es.wikipedia.org/wiki/Factor_Rh
En general, las personas que no somos ilustradas en el tema, decimos que el tipo de sangre es A+ u O-, por ejemplo, sin que sea preciso. Haciendo caso omiso del concepto real, **suponga que se tiene una lista** con los siguientes datos de varios estudiantes:

Carné, tipo de sangre y sexo. El tipo de sangre es O+, A+, B+, AB+, O-, A-, B- y AB-. El sexo será f (femenino) o m (masculino). Por ejemplo:

```
datos_est = [["3567", "A+", 'm'], ["1122", "A+", 'f'], ["4432", "O+", 'f'], ["8734", "O-", 'm'], ["1290", "B+", 'f'], ["6512", "A+", 'm'], ["6523", "A-", 'm'], ["9008", "B-", 'm'], ["8745", "A+", 'f'], ["3131", "A-", 'm'], ["8811", "A+", 'f']]
```

Construya **una función** en Python que inicialice una lista con todos los datos de los estudiantes de cierto sexo y cierto tipo de sangre. La misma función debe servir para cualquier combinación de esos dos datos que se especifiquen como argumentos.

La lista debe quedar disponible donde se invoque la función.

3. **Suponga que se tienen dos listas (es decir, ya existen).** Una con código del producto, nombre del producto, proveedor, descuento y precio por kilo, y otra con código y nombre del proveedor. Por ejemplo:

```
datos_prod = [["1234", "lenteja", ["1516", 30, 4000]],  
              ["9831", "queso", ["1516", 10, 12000]],  
              ["3358", "carne", ["9128", 20, 32000]],  
              ["9166", "naranja", ["9128", 20, 800]],  
              ["1132", "arroz", ["6427", 20, 2000]],  
              ["1772", "guanábana", ["6427", 30, 5500]],  
              ["9166", "naranja", ["6427", 10, 800]],  
              ["4819", "tomate", ["9128", 10, 6500]],  
              ["1234", "lenteja", ["9128", 10, 4000]],  
              ["3358", "carne", ["6427", 0, 32000]],  
              ["3388", "curuba", ["6427", 10, 3000]],  
              ["9922", "habichuela", ["9128", 20, 1500]],
```

```
["3443", "cebolla cabezona", ["9128", 0, 2300]],
["1114", "cilantro", ["1516", 10, 1200]],
["1772", "guanábana", ["1516", 0, 5500]]
```

```
datos_provee = ["1516", "Agricultores Unidos",
                ["9128", "Campesinos del Norte"],
                ["6427", "Del Campo a su Mesa"]]
```

Construya **una función** en Python que agregue una lista a la segunda lista con los productos de cada proveedor, los cuales están en la primera lista.

La lista modificada debe quedar disponible donde se invoque la función. **Con los datos del ejemplo**, la segunda lista quedaría así:

```
datos_provee = ["1516", "Agricultores Unidos", ["lenteja", "queso", "cilantro", "guanábana"],
                ["9128", "Campesinos del Norte", ["carne", "naranja", "tomate", "lenteja", "habichuela", "cebolla cabezona"]],
                ["6427", "Del Campo a su Mesa", ["arroz", "guanábana", "naranja", "carne", "curuba"]]]
```

4. **Suponga que se tiene una lista** de estudiantes con nombre y calificaciones de cada tercio en una asignatura. Por ejemplo:

```
estudiantes = ["María Pérez", [4.3, 5.0, 4.5]],
               ["Simón Durán", [4.0, 3.8, 3.5]],
               ["María Pérez", [2.8, 3.5, 4.0]],
               ["Ricardo Lozano", [2.0, 2.3, 2.5]],
               ["Ángela Matiz", [3.2, 2.9, 3.3]],
               ["Claudia Pinzón", [4.5, 4.5, 5.0]],
               ["Silvia Urbina", [1.9, 1.7, 0.0]],
               ["Gonzalo Quintana", [3.0, 2.9, 3.4]]]
```

Construya una función en Python que guarde en otra lista los estudiantes y la definitiva, siempre y cuando esta última esté en un rango que también será enviado como argumento. Para el cálculo de la definitiva hay que tener en cuenta que las dos primeras calificaciones valen 30 %, cada una, y la tercera, 40 %.

La lista debe quedar disponible en la función que invoque a ésta. Los datos pueden ser otros, con la misma estructura. **Con los datos del ejemplo**, si el rango fuera [1.0, 2.9], la lista quedaría así:

```
[["Ricardo Lozano", 2.3], ["Silvia Urbina", 1.1]]
```

5. **Suponga que se tienen dos listas (es decir, ya existen)**. Una con código del producto, nombre del producto, proveedor y descuento, y otra con código del producto y precio por kilo. Por ejemplo:

```
datos_prod = [
    ["1234", "lenteja", ["Agricultores Unidos", 30]],
    ["9831", "queso", ["Agricultores Unidos", 10]],
    ["3358", "carne", ["Campesinos del Norte", 20]],
    ["9166", "naranja", ["Campesinos del Norte", 20]],
    ["1132", "arroz", ["Del Campo a su Mesa", 20]],
    ["1772", "guanábana", ["Del Campo a su Mesa", 30]],
    ["9166", "naranja", ["Del Campo a su Mesa", 10]],
    ["4819", "tomate", ["Campesinos del Norte", 10]],
    ["1234", "lenteja", ["Campesinos del Norte", 10]],
    ["3358", "carne", ["Del Campo a su Mesa", 0]],
    ["3388", "curuba", ["Del Campo a su Mesa", 10]],
    ["9922", "habichuela", ["Campesinos del Norte", 20]],
    ["3443", "cebolla cabezona", ["Campesinos del Norte", 0]],
    ["1114", "cilantro", ["Agricultores Unidos", 10]],
    ["1772", "guanábana", ["Agricultores Unidos", 0]]
]
```

```
cod_prod_precio = [
    ["9831", 12000],
    ["1132", 2000],
    ["9166", 800],
    ["4819", 6500],
    ["3358", 32000],
    ["9922", 1500],
    ["1234", 4000],
    ["3443", 2300],
    ["1114", 1200],
    ["1772", 5500],
    ["3388", 3000]
]
```

Construya una función en Python que agregue a la primera lista el precio de cada producto, el cual está en la segunda lista.

La lista modificada debe quedar disponible en la función que invoque a ésta. **Con los datos del ejemplo**, la primera lista quedaría así:

```
datos_prod = [["1234", "lenteja", ["Agricultores Unidos", 30, 4000]],
              ["9831", "queso", ["Agricultores Unidos", 10, 12000]],
              ["3358", "carne", ["Campesinos del Norte", 20, 32000]],
              ["9166", "naranja", ["Campesinos del Norte", 20, 800]],
              ["1132", "arroz", ["Del Campo a su Mesa", 20, 2000]],
              ["1772", "guanábana", ["Del Campo a su Mesa", 30, 5500]],
              ["9166", "naranja", ["Del Campo a su Mesa", 10, 800]],
              ["4819", "tomate", ["Campesinos del Norte", 10, 6500]],
              ["1234", "lenteja", ["Campesinos del Norte", 10, 4000]],
              ["3358", "carne", ["Del Campo a su Mesa", 0, 32000]],
              ["3388", "curuba", ["Del Campo a su Mesa", 10, 3000]],
              ["9922", "habichuela", ["Campesinos del Norte", 20, 1500]],
              ["3443", "cebolla cabezona", ["Campesinos del Norte", 0, 2300]],
              ["1114", "cilantro", ["Agricultores Unidos", 10, 1200]],
              ["1772", "guanábana", ["Agricultores Unidos", 0, 5500]]]
```

6. Construyan un **programa modular** para crear y escribir una lista de jugadores de baloncesto y cuyos registros tengan la siguiente estructura:

[apellido, nombre, [estatura en metros, peso en kilogramos]]

6.1 Cree la lista.

6.2 Pida y valide la cantidad de jugadores.

6.3 Pida los datos de los jugadores.

6.4 Calcule el Índice de Masa Corporal (IMC) y agréguelo a la lista. $IMC = \text{peso en kg} / (\text{estatura en m})^2$

[apellido, nombre, [estatura en metros, peso en kilogramos, imc]]

Ejemplo: ["Segura", "Sonia", [1.70, 75.2, 26.02]]

6.5 Escriba un reporte enumerado.

Cada uno de los puntos se debe desarrollar en una función. Les doy el esqueleto.

Sugerencia: Reutilice todo lo que sea posible y que hayamos trabajado en el curso.