UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CS1111 Programación I
Tarea 4
Entrega mediante: Gradescope

Indicaciones generales

- Recuerda que la **tarea es individual**. Los casos de copia/plagio serán sancionados con nota cero (0) en la tarea.
- La tarea será publicada en Canvas pero **se entrega por la plataforma Gradescope**, además debe considerar:
  - Se le proporcionará un archivo solution.py. Usted deberá editarlo en las secciones correspondientes.
  - Al finalizar, solo subir el archivo solution.py (**NO cambiar el nombre del archivo y NO comprimirlo**).
  - Cada pregunta tiene casos de prueba, evalue su solución con cada caso.
  - El Gradescope evaluará los ejercicios con entradas diferentes a las que se muestran en este documento. Se recomienda probar con sus propias entradas.
  - Las consultas y reclamos serán atendidos durante los horarios de asesorías.

Gradescope

1. Nosotros les proporcionaremos un código base de donde deberán partir para completar dicho ejercicio. Este archivo es llamado `solution.py` y lo encontrarán en la indicación de la tarea en CANVAS.
2. Al finalizar, **solo** subir el archivo `solution.py` (NO cambiar el nombre del archivo y NO comprimirlo).
3. Cada pregunta tiene diversos casos de prueba. Para obtener la nota completa en una pregunta, el algoritmo debe obtener la respuesta correcta en dichos casos de prueba.
4. Si un caso de prueba falla, visualizarán un mensaje de error con sugerencias. **Lee el error**, revisa el código e inténtalo de nuevo.
5. Los input de los casos de prueba son confidenciales.

Indicaciones específicas

1. Ustedes deben escribir dentro de la sección y a la misma altura de donde esta escrito *"SU SOLUCION EMPIEZA AQUI"*. Además, no deben modificar nada debajo de *"SU SOLUCION TERMINA AQUI"*. Recuerden tener cuidado con las indentaciones.

2. NO SE PERMITE USAR LA FUNCION **sort** ni **sorted** de python.

1 Problema 1: ordenando tareas (5pts)

Usted es un empleado de una empresa que busca elaborar un algoritmo capaz de ordenar las tareas que se tienen a partir de las diferentes características de la tarea. El algoritmo recibirá una lista con las tareas, cada una en formato de diccionario; el valor a ordenar; y si se desea ordenar de forma ascendente o descendente.

1.1 Ejemplo de formato de tarea

```
tareas = [
{
    'id': 1,
    'descripcion': 'Tarea 1',
    'importancia' : 1
},
{
    'id': 2,
    'descripcion': 'Tarea 2',
    'importancia' : 2
},
]
```

No se requiere utilizar inputs, asumir que los valores que recibe la función ya recibe los datos solicitados. La función debe retornar la lista de diccionarios ordenado.
NOTA: La funcion interpreta el tipo de ordenamiento de la siguiente forma: **ascendente:** 'asc' y **descendente:** 'desc'

1.2 Ejemplo de input-output

Input

```
pregunta1(
        [{'id': 6, 'descripcion': 'Tarea 6', 'importancia': 6},
        {'id': 2, 'descripcion': 'Tarea 2', 'importancia': 2},
        {'id': 5, 'descripcion': 'Tarea 5', 'importancia': 5},
        {'id': 4, 'descripcion': 'Tarea 4', 'importancia': 4},
        {'id': 3, 'descripcion': 'Tarea 3', 'importancia': 3},
        {'id': 1, 'descripcion': 'Tarea 1', 'importancia': 1},
        ], "id", "asc")
```

Output

```
[{'id ': 1 , 'descripcion': 'Tarea 1' , 'importancia' : 1},
{'id ': 2 , 'descripcion': 'Tarea 2' , 'importancia' : 2},
{'id ': 3 , 'descripcion': 'Tarea 3' , 'importancia' : 3},
{'id ': 4 , 'descripcion': 'Tarea 4' , 'importancia' : 4},
{'id ': 5 , 'descripcion': 'Tarea 5' , 'importancia' : 5},
```

```
    {'id ': 6 , 'descripcion': 'Tarea 6' , 'importancia' : 6},]
```

2 Ejercicio 2: Tareas específicas (5pts)

La misma empresa, te ha solicitado ahora un algoritmo de búsqueda para poder saber si existe o no la tarea x. Para esta ocación te solicitan hacer una función recursiva de búsqueda.

NOTA : Asumir que el input es una lista de diccionarios ordenada de tareas por su id y la búsqueda se realizará por el valor del id.

2.1 Ejemplo de formato de diccionario

```
    tareas = [
    {
        'id': 1,
        'descripción': 'Tarea 1',
        'importancia' : 1
    },
    {
        'id': 2,
        'descripción': 'Tarea 2',
        'importancia' : 2
    }
    ]
```

2.2 Ejemplo de input-output

Input

```
pregunta2([
    {
        'id': 1,
        'descripción': 'Tarea 1',
        'importancia' : 1
    },
    {
        'id': 2,
        'descripción': 'Tarea 2',
        'importancia' : 2
    }
    ], 1)
```

Output

```
    {'id': 1,
    'descripción': 'Tarea 1',
```

```
    'importancia' : 1}
```

3 Ejercicio 3: Sistema de gestión de inventario de una librería en línea (10pts)

Se te ha asignado la tarea de desarrollar un sistema de gestión de inventario para una librería en línea. El sistema debe manejar varias operaciones relacionadas con el inventario de libros, incluyendo la búsqueda de libros, la clasificación de libros y la gestión de la información de los libros utilizando diccionarios.

El inventario se almacena como una lista de diccionarios que contienen información del libro, como título, autor, precio y cantidad disponible. Aquí tienes un ejemplo de representación de un libro en el diccionario de inventario:

```
inventario = [
{
    'titulo': 'El Gran Gatsby',
    'autor': 'F. Scott Fitzgerald',
    'precio': 12.99,
    'cantidad': 5
},
{
    'titulo': 'Matar a un ruisenor',
    'autor': 'Harper Lee',
    'precio': 9.99,
    'cantidad': 8
},
# ... más libros
]
```

Debes implementar las siguientes funcionalidades:

- **Busqueda**: El sistema debe permitir a los usuarios buscar libros por título o autor. Dada una consulta de búsqueda, el sistema debe recorrer el diccionario de inventario y devolver una lista de libros que coincidan exactamente con la consulta de búsqueda. Coincidencias parciales no se toman en consideracion.

- **Clasificacion**: El sistema debe proporcionar opciones de clasificación para mostrar los libros. Los usuarios deben poder ordenar los libros por título o precio en orden ascendente o descendente. El algoritmo de clasificación debe utilizar recursividad para lograr el orden de clasificación deseado.

NOTA 1: La funcion de busqueda es realizada sobre un arreglo de diccionarios desordenada.
NOTA 2: Al llamar a la funcion *pregunta_3* se envian los siguientes parametros respetando el orden Diccionario, tipo de funcionalidad, nombre de columna, valor a buscar, tipo de ordenamiento. Cuando se desea realizar la funcionalidad de **Clasificacion** el parametro de *valor*

se ingresa como comillas vacias. Mientras que al llamar a la funcion en modo de **Busqueda** el parametro de *ord_type* es en comillas simples.

3.1 Ejemplo input-output

Input

```
pregunta3([{'titulo': 'El Gran Gatsby',
        'autor': 'F. Scott Fitzgerald',
        'precio': 12.99,
        'cantidad': 5},
        {'titulo': 'Matar a un ruisenor',
        'autor': 'Harper Lee',
        'precio': 9.99,
        'cantidad': 8},
        {'titulo': 'Cronica de una muerte anunciada',
        'autor': 'Gabriel Garcia Marquez',
        'precio': 9.45,
        'cantidad': 10},
        {'titulo': 'La lluvia sabe por que',
        'autor': 'María Fernanda Heredia',
        'precio': 10,
        'cantidad': 4}], 'Clasificacion', 'titulo', '', 'asc')
```

Output

```
[{
        'titulo': 'Cronica de una muerte anunciada',
        'autor': 'Gabriel Garcia Marquez',
        'precio': 9.45,
        'cantidad': 10,
    },
    {
        'titulo': 'El Gran Gatsby',
        'autor': 'F. Scott Fitzgerald',
        'precio': 12.99,
        'cantidad': 5
    },
    {
        'titulo': 'La lluvia sabe por que',
        'autor': 'María Fernanda Heredia'
        'precio': 10,
        'cantidad': 4
    },
    {
        'titulo': 'Matar a un ruisenor',
        'autor': 'Harper Lee',
```

```
        'precio': 9.99,
        'cantidad': 8}]
```

Input

```
pregunta3([{'titulo': 'El Gran Gatsby',
        'autor': 'F. Scott Fitzgerald',
        'precio': 12.99,
        'cantidad': 5},
        {'titulo': 'Matar a un ruisenor',
        'autor': 'Harper Lee',
        'precio': 9.99,
        'cantidad': 8},
        {'titulo': 'Cronica de una muerte anunciada',
        'autor': 'Gabriel Garcia Marquez',
        'precio': 9.45,
        'cantidad': 10},
        {'titulo': 'La lluvia sabe por que',
        'autor': 'María Fernanda Heredia',
        'precio': 10,
        'cantidad': 4}], 'Busqueda', 'autor', 'Gabriel Garcia
   Marquez', '')
```

Output

```
    {
        'titulo': 'Cronica de una muerte anunciada',
        'autor': 'Gabriel Garcia Marquez',
        'precio': 9.45,
        'cantidad': 10,
    }
```