

Algoritmos y Estructuras de Datos
Examen Final 30/11/2023 – Regulares

Enunciado General:

Una Compañía Aérea requiere un programa para gestionar los datos de los tickets que tiene vendidos a sus pasajeros. Por cada **ticket** se tienen los siguientes datos: código del vuelo (una cadena), nombre del pasajero (una cadena), un número entero entre 1 y 140 para indicar el número de asiento asignado, otro número entero (pero entre 0 y 9) para indicar la clase del pasaje (por ejemplo: 0: Turista, 1: Económica, 2: Ejecutiva, etc.), el aeropuerto de origen (una cadena), el aeropuerto de destino (una cadena) y finalmente un número en coma flotante para indicar el monto que se percibió por ese ticket.

En base a lo anterior, se provee **ya desarrollado**

- ✓ Un módulo **soporte.py** que contiene la definición de la clase **Ticket** con el **método constructor** y el método **__str__()** ya incluidos.
- ✓ Un segundo módulo **principal.py** que contiene una función **main()** con el esquema del menú de opciones ya desarrollado y funcionando, de forma que esa función **main()** sea usada como la función de entrada del programa completo. El módulo **principal.py** contiene **ya desarrollados e incluidos en el menú** algunos procesos y funciones, según se indica a continuación:
 - **Opción 1 del menú (ya desarrollada):** Generar un arreglo de **n** objetos de tipo **Ticket** que contenga los datos de todos los tickets. La carga del valor de **n** por teclado validando que sea correcto está ya desarrollada también. El arreglo se genera con datos aleatorios, en forma automática, eliminando los datos anteriores cada vez que se ejecuta la opción 1, y se mantiene ordenado por código de vuelo en todo momento (la función **add_in_order()** está ya desarrollada y provista en el mismo módulo **principal.py**).
 - **Opción 2 del menú (ya desarrollada):** Mostrar todos los datos del arreglo, a razón de un registro por renglón.

SU TAREA: Desarrolle funciones para cumplir con los siguientes requerimientos **[Máximo 4 puntos por convenciones de estilo y la estructura de las funciones que desarrolle]**:

1. **Opción 3 del menú:** Determinar si existe en el arreglo un objeto cuyo código de vuelo sea igual a **cod** (cargar **cod** por teclado, sin exigencia de validación). Si existe, modifique ese objeto de modo que se sume al monto del pasaje un 10% de su valor anterior, y muestre el objeto cambiado. Si no existe, informe con un mensaje. La búsqueda debe detenerse al encontrar el primer objeto que cumpla el criterio de búsqueda pedido. **[Máximo 4 puntos]**.
2. **Opción 4 del menú:** Determinar la cantidad de tickets vendidos para cada combinación entre número de asiento y clase del pasaje (un contador para los que sean asiento 1 y clase 0, otro para el asiento 1 y clase 1, y así sucesivamente para las $140 \times 10 = 1400$ combinaciones posibles). Mostrar solo los conteos diferentes de cero. **[Máximo 4 puntos]**.
3. **Opción 5 del menú:** Grabar en un archivo binario los datos de los objetos del arreglo que correspondan a tickets cuyo aeropuerto de origen sea igual a **aer1** (cargue la variable **aer1** por teclado) y cuyo aeropuerto de destino sea igual a **aer2** (cargue por teclado la variable **aer2**), pero en los que además la clase del pasaje NO sea ni 0 ni 1. Aclaración: para crear el archivo pedido, NO DEBE CREAR un segundo arreglo. **[Máximo 4 puntos]**.
4. **Opción 6 del menú:** Mostrar el archivo generado en el anterior a razón de una línea en pantalla por cada registro. Muestre al final dos líneas extra, indicando la cantidad de registros que se mostraron, y el monto promedio pagado entre todos los tickets que se mostraron. **[Máximo 4 puntos]**.

Criterios generales de corrección: La suma total de puntos llega a un máximo de 20 (considerando los 4 puntos por convenciones de estilo y estructura de las funciones presentadas). Para aprobar necesita llegar a 12 puntos (el 60% de ese puntaje máximo).