Consigna 1

Una aerolínea de bajo costo cobra un adicional sobre el precio *del* pasaje a quienes deseen elegir asiento.

Quienes no lo abonen serán asignados al azar en los asientos que no hayan sido elegidos previamente.

Esta información se almacena por cada vuelo en un archivo CSV donde cada registro puede tener dos formatos diferentes:

Formato 1:

Nombre *del* pasajero; fila; asiento

Formato 2:

Nombre *del* pasajero

El formato 1 se aplica a quienes hayan abonado el adicional por seleccionar ubicación, mientras que el formato 2 se aplica a los demás viajeros.

El nombre *del* pasajero se escribe como «Apellido, nombre». La fila oscila entre 1 y N, donde el valor de N depende *del* modelo de la aeronave.

El asiento va desde la A hasta la F en modelos regionales de hasta 32 filas de asientos y desde la A hasta la J en modelos de doble pasillo y

largo alcance con más de 32 filas de asientos.

Se solicita desarrollar un programa que lea un archivo con las características mencionadas y distribuya los asientos respetando las reservas

realizadas y asignando al azar los asientos restantes entre los pasajeros que no abonaron el adicional. Mostrar un listado con las asignaciones

realizadas ordenado por fila y asiento, e imprimir otro listado con los asientos que hayan quedado libres ordenado por el mismo criterio.

También se deberá emitir un listado con las colisiones, si las hubiera. Una colisión ocurre cuando dos pasajeros distintos reservan el mismo asiento.

Estos eventos deben ser resueltos por personal de la aerolínea.

Se suministra un archivo ejemplo llamado Vuelo447.txt. El archivo no está ordenado por ningún criterio particular.

El programa debe funcionar con éste o cualquier otro archivo que respete las características indicadas, detectando

automáticamente el modelo de avión (simple o doble pasillo) en función de la cantidad de filas de la aeronave y de los asientos reservados.

Es decir que si todas las reservas tienen una fila menor o igual a 32 y todos los asientos solo poseen letras de la A a la F, deberá

considerarse que se trata de un avión de un solo pasillo y alcance regional.

final_aerolinea_example.py

Consigna 2

Se tienen dos archivos con los datos de los alumnos inscriptos para rendir un examen.

Cada registro contiene:

Número de legajo

Nombre

Turno (M: Mañana; T: Tarde; N: Noche)

Existe un registro por cada alumno. Los archivos se encuentran ordenados por número de legajo.

Obtener un archivo único con los inscriptos para el examen en el que conste la misma información, el que también debe quedar ordenado por número de legajo.

Debido a errores en la carga de datos puede haber alumnos que aparezcan inscriptos en más de un turno.

Esos alumnos deberán mostrarse por pantalla en un listado de inconsistencias, y no se grabarán en el archivo de salida.

Al finalizar el proceso informar la cantidad de registros grabados y la cantidad de inconsistencias halladas.

Se adjuntan dos archivos llamados alumnos1.txt y alumnos2.txt, ambos con formato CSV (valores separados por punto y coma).

Los mismos han sido creados con codificación UTF8, por lo que deberá agregarse el parámetro encoding="UTF8" en el momento de su apertura.

El programa debe servir para estos archivos o cualquier otro que respete el mismo formato.

Consigna 3

Se solicita leer un archivo de texto cualquiera e informar cuántas veces aparece cada palabra en el mismo,

sólo para aquellas palabras que tengan tres o más letras. Tener en cuenta que:

a. El archivo puede leerse una sola vez.

b. No debe distinguirse entre mayúsculas y minúsculas, es decir que “Hola” y “hola” son la misma palabra.

c. Todos los números y signos de puntuación deben ser ignorados. No limitarlo a un conjunto de símbolos.

d. Mostrar por pantalla las palabras y sus repeticiones ordenadas de menor a mayor por longitud, y dentro de

cada longitud en orden alfabético.

e. Mostrar a las palabras que contengan la mayor cantidad de vocales, sin importar si tienen o no tilde o diéresis.

f. Se suministra un archivo de ejemplo llamado “Historia.txt”. El programa debe funcionar con

éste o cualquier otro archivo de texto.