 Deusto Facultad de Ingeniería Ingeniaritza Fakultatea	EXAMEN DE PROGRAMACIÓN I	7-FEBRERO-2022
---	-----------------------------	----------------

Nombre: _____

DNI: _____ Aula: _____ Fila: _____ Columna _____

Instrucciones

- El examen se recogerá en pendrive al terminar.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Guarda el código de todos los ejercicios en un único fichero que se llame “ExamenFinal-NombreApellidos.py”, por ejemplo: ExamenFinal-PabloGaraizar.py.

Enunciado

DeustoStudent es un nuevo servicio de la universidad para poder gestionar *online* la lista de asignaturas de los estudiantes de la Universidad de Deusto. Te han contratado para crear una aplicación que les ayude a gestionar sus asignaturas y los estudiantes:

Asignatura

Cada asignatura requiere almacenar la siguiente información:

- **id:** cadena de caracteres (str), por ejemplo “FI123”.
- **nombre:** cadena de caracteres (str), por ejemplo “Programación I”.
- **nota:** (float), entre 0 y 10.
- **convocatorias:** (int), de 0 a 6.

Estudiante


Cada estudiante de DeustoStudent requiere almacenar la siguiente información:

- **id:** código numérico (int), de 1 en adelante.
- **usuario:** cadena de caracteres (str), por ejemplo “garaizar”.
- **población:** cadena de caracteres (str), por ejemplo “Amoroto”.
- **asignaturas:** lista de asignaturas de este estudiante.

Modela asignaturas y estudiantes como diccionarios o listas multidimensionales.

Crea un programa con una lista de asignaturas vacía, una lista de estudiantes vacía y programa las siguientes funciones:

asignaturaStr: función que recibe una asignatura y devuelve una cadena de caracteres (str) con el siguiente formato: ‘FI001;Programación I, 7, 1’ **(0.5 puntos)**

 Deusto Facultad de Ingeniería Ingeniaritza Fakultatea	EXAMEN DE PROGRAMACIÓN I	7-FEBRERO-2022
---	-----------------------------	----------------

cargarAsignaturas: función que recibe la lista de asignaturas vacía y añade a la lista las asignaturas almacenados en el fichero 'asignaturas.csv' que tiene el siguiente formato: **id;nombre (2 puntos)**. Por ejemplo:

FI001;Programación I

Atención: si no sabes hacer esta función, sustitúyela por la creación a mano de una lista de 10 asignaturas. Esta solución alternativa valdrá 0.5 puntos.

crearEstudiantes: función que recibe la lista de asignaturas llena y la lista de estudiantes vacía y devuelve una lista de estudiantes llena. Para ello, solicitar los datos del estudiante (nombre y población) y mediante un menú que muestre todas las asignaturas, añadir tantas asignaturas como fuera necesario, verificando que la asignatura no está duplicada comparando el código. También debe de introducirse la nota y el número de convocatorias del estudiante para esa asignatura **(2 puntos)**

Atención: si no sabes hacer esta función, sustitúyela por la creación a mano de una lista de 10 asignaturas. Esta solución alternativa valdrá 0.5 puntos.

estudiantesMatriculados: función que recibe el id de una asignatura y una lista de estudiantes y devuelve el número total de estudiantes que están matriculados de esa asignatura **(1 punto)**

asignaturaConMasSuspendos: función que recibe la lista de estudiantes y devuelve el nombre de la asignatura con más suspendos. **(2 puntos)**

numeroEstudiantesPoblación: función que recibe la lista de estudiantes y devuelve el un diccionario cuya clave es el nombre de la población y el valor el número de estudiantes que hay en la lista de esa población. **(2 puntos)**

Como se ha comentado anteriormente, además de estas funciones, desarrolla un **programa principal** en el que haya una lista de **asignaturas** vacía (que llenarás con la función **cargarAsignaturas**) y una lista de **estudiantes** vacía (que llenarás con la función **crearEstudiantes**) y se llame a las funciones **asignaturaStr**, **alumnosMatriculados**, **asignaturaConMasSuspendos**, **numeroEstudiantesPoblación**. **(0.5 puntos)**