

## ESTRUCTURAS DE DATOS Y ALGORITMOS

2º de Grado en Ingeniería Matemática (Curso 2021-2022)

### PRÁCTICA 2

#### 1. NORMAS GENERALES

1. El código deberá mostrar **modularidad, calidad, legibilidad y el uso apropiado de estructuras y la inclusión de comentarios**. En esta práctica podrá codificarse tanto en **C (opción preferida)** como en Python.
2. El alumno deberá subir a la plataforma virtual la práctica realizada. La subida al espacio habilitado podrá realizarse hasta el **28/11/2021** en convocatoria ordinaria y hasta el 12/06/2021 en convocatoria extraordinaria, ambas fechas inclusive.
3. Se deberá subir únicamente un fichero \*.rar, nombrado del siguiente modo: PR2\_NombreAlumno1Apellido1Alumno1NombreAlumno2Apellido1Alumno2.rar (Ejemplo: PR2\_AntonioPerezLuisFernandez.rar). El nombre y el apellido del alumno no deberá contener acentos.
4. El fichero \*.rar contendrá:
  - Los ficheros fuente, nombrados siguiente modo:  
PR2\_P1\_NombreAlumno1Apellido1Alumno1NombreAlumno2Apellido1Alumno2.c  
(Ejemplo: PR2\_P1\_AntonioPerezLuisFernandez.c,  
PR2\_P2\_AntonioPerezLuisFernandez.c). El nombre y apellido de los alumnos no deberán contener acentos.
  - En el caso Python los .py.
5. Aquellos ejercicios cuyos ficheros de entrega no cumplan la normativa de nombrado se calificarán con 0 puntos.
6. El ejercicio deberá realizarse en equipo (pareja de prácticas)

#### 2. OBJETIVOS

- El objetivo es familiarizarse con algunas estructuras de datos: conjuntos, pilas, listas y colas.

### 3. ENUNCIADO

#### 3.1. Primera parte: Conjuntos

Debe desarrollarse un programa en el que dado un archivo (fich01.txt, lo proporciona el profesor) que contiene en cada línea nombres de ciudades cree tres conjuntos. El conjunto 1 que contiene las ciudades del archivo con un número de caracteres menor o igual a cuatro, el conjunto 2 que alberga las ciudades del archivo con un número de caracteres  $\leq 6$  y finalmente el conjunto 3, que tiene las ciudades que poseen un número de caracteres  $> 4$ . El programa debe realizar las siguientes operaciones con los conjuntos.

Diseñar y codificar un programa que muestre por la salida estándar un menú con las siguientes opciones:

##### **MENU**

1. Leer fichero.
2. Mostrar conjuntos 1 y 2.
3. Unir conjuntos 1 y 2.
4. Intersección de conjuntos 1 y 2.
5. Diferencia de conjuntos 1 y 2.
6. Comprobar cadena.
7. Salir del programa.

A continuación, se explican cada una de las opciones:

1. Leer fichero. Lee el fichero con los datos de entrada.
2. Mostrar conjuntos 1 y 2. Muestra en pantalla el contenido de cada conjunto.
3. Unir conjuntos 1 y 2. Realiza la unión de conjuntos y muestra el resultado en pantalla.
4. Intersección de conjuntos 1 y 2. Realiza la intersección de conjuntos y muestra el resultado en pantalla.
5. Diferencia de conjuntos 1 y 2. Realiza la diferencia de conjuntos y muestra el resultado en pantalla.
6. Comprobar cadena. Comprobará si la cadena de caracteres introducida por pantalla se encuentra en alguno de los conjuntos.
7. Salir. Se mostrará la palabra "Adios" y el programa terminará.

## 3.2.Segunda parte: Pilas y Colas

Diseñar y codificar un programa que muestre por la salida estándar un menú con las siguientes opciones:

### **MENU**

1. Gestionar pilas.
2. Mostrar pilas.
3. Invertir pilas.
4. Gestionar colas.
5. Mostrar colas.
6. Invertir colas.
7. Salir.

Se trabajará con un total de 3 pilas ( $n=3$ ). La pila 1 tiene como elementos cadenas de caracteres, la 2 números enteros y la 3 una estructura. La cual guarda la siguiente información sobre la cadena introducida por pantalla: número de caracteres.

### **1. Gestionar pilas**

Se solicitará al usuario que introduzca una cadena de caracteres, que contiene grupos de caracteres separados por espacios en blanco. La cadena es de la forma:

`<grupo 1> <grupo2> <grupo3> <grupo4>...<grupo n>`

Se procederá de la forma siguiente, con cada grupo:

Si un grupo contiene solo letras se introducirá en la pila 1, si tiene solo números se introducirá en la pila 2. Si tiene letras y números no se introducirá en ninguna pila. En la pila 3 se guardará la siguiente información sobre la cadena introducida por pantalla: número de caracteres, longitud.

Debe respetarse el mecanismo de gestión de pilas (accediendo por la cima).

### **2. Mostrar pilas**

Se mostrará el contenido de las tres pilas en pantalla.

### **3. Invertir pilas**

Se invertirán las tres pilas.

### **4. Gestionar colas**

Se solicitará al usuario que introduzca una cadena de caracteres, que contiene grupos de caracteres separados por espacios en blanco. La cadena es de la forma:

`<grupo 1> <grupo2> <grupo3> <grupo4>...<grupo n>`

Se procederá de la forma siguiente, con cada grupo:

Si un grupo contiene solo letras se introducirá en la cola 1, si tiene solo números se introducirá en la cola 2. Si tiene letras y números no se introducirá en ninguna cola. En la cola 3 se guardará la siguiente información sobre la cadena introducida por pantalla: número de caracteres, longitud.

Debe respetarse el mecanismo de gestión de colas (introduciendo elementos al final).

### 5. **Mostrar colas**

Se mostrará el contenido de las tres pilas en pantalla.

### 6. **Invertir colas**

Se invertirán las tres colas.

### 7. **Salir**

Se mostrará la palabra “Adios” y el programa terminará.

## 3.3. Tercera parte: listas enlazadas

Escriba un programa que gestione el inventario de una PYME. La información debe ser almacenada en una lista enlazada. La lista tendrá un elemento, por cada artículo diferente existente. Cada elemento de la lista contendrá: identificador del artículo, la cantidad actualmente existente y la mínima requerida.

El programa mostrará el siguiente menú:

#### **MENU**

1. Registrar artículo nuevo
2. Eliminar artículo
3. Registrar llegadas al almacén
4. Registrar salidas del almacén
5. Mostrar el inventario
6. Salir del programa

A continuación se explican cada una de las opciones.

1. **Registrar artículo nuevo.** Se introducirá una cadena de caracteres con el siguiente formato (cualquier otro formato se calificará con cero puntos).

<Identificador de artículo> <Cantidad en almacén> <Cantidad mínima>

Ejemplo:

123456 45 12

**OPCIONAL:** Cada vez que se introduzca un artículo la lista debe permanecer

ordenada según el identificador del artículo. Indicar claramente en los comentarios del principio del fichero si se ha implementado este mecanismo de inserción ordenada, que es **opcional**.

2. **Eliminar artículo.** Se introducirá una cadena de caracteres con el siguiente formato (cualquier otro formato se calificará con cero puntos).

<Identificador de artículo>

Ejemplo:

123456

3. **Registrar llegadas al almacén**

<Identificador de artículo> <Unidades>

Ejemplo:

123456 15

4. **Registrar salidas del almacén**

<Identificador de artículo> <Unidades>

Ejemplo:

123456 15

5. **Mostrar inventario**

Se mostrará el inventario ordenado por su identificador.

6. **Salir del programa**

Escribe “Adios” y el programa termina.