

ESTRUCTURAS DE DATOS Y ALGORITMOS

2º de Grado en Ingeniería Matemática (Curso 2022-2023)

PRÁCTICA 1

1. NORMAS GENERALES

1. El código deberá mostrar modularidad, calidad, legibilidad y el uso apropiado de estructuras y la inclusión de comentarios.
2. Esta práctica deberá realizarse en C. Deberá respetarse el menú de opciones.
3. El alumno deberá subir a la plataforma virtual la práctica realizada. La subida al espacio habilitado podrá realizarse hasta el 06/11/2022 en convocatoria ordinaria y hasta el 04//06/2023 en convocatoria extraordinaria, ambas fechas inclusive.
4. Se deberá subir únicamente un fichero ***.rar / *.zip**, nombrado del siguiente modo: PR1_Nom1Apell1_Nom2Apell2_Nomb3_Apell3.rar (Ejemplo: PR1_AntonioPerez_LuisaRamirez_LaraAlvarez.rar). El nombre y el apellido del alumno no deberá contener acentos.
5. El fichero *.rar / *.zip contendrá únicamente los ficheros fuente, nombrados siguiente modo: PR1_P1_Nom1_Nom2_Nom3.c, PR1_P2_Nom1_Nom2_Nom3.c (Ejemplo: PR1_P1_Antonio_Luisa_Lara.c, ...). El nombre y apellidos del alumno, no deberán contener acentos.
6. Aquellos ejercicios cuyos ficheros de entrega no cumplan la normativa de nombrado se calificarán con 0 puntos.
7. El ejercicio deberá realizarse en equipo (3 personas).

2. OBJETIVOS

- El objetivo es familiarizarse con el manejo de memoria dinámica, entendiendo sus ventajas frente a la utilización de memoria estática. También el uso y manejo de ficheros y las funciones de manipulación de strings.

3. ENUNCIADO

3.1 Programa 1

Se implementarán las siguientes opciones de menú:

MENU

1. Leer información de entrada.
2. Eliminar línea.
3. Mostrar línea.
4. Mostrar datos de usuario.
5. Salir del programa.

Se describe a continuación la funcionalidad de cada una de estas opciones:

1. Leer información de entrada

Lee un fichero, `fich01.csv`, que contiene un listado de personas, el cual incluye el nombre, apellido, ciudad y país donde reside. La información de cada línea es:

`<Nombre>;<Apellido>;<País>;<Ciudad>`

La información debe ser almacenada en memoria dinámica.

2. Eliminar línea.

(i)Pide el número de línea a eliminar, (ii) borra la información correspondiente a la línea indicada de la memoria, (iii) guarda la información resultante tras realizar la operación en el fichero `fich01.csv`.

3. Mostar línea.

(i)Pide el número de línea a mostrar, (2) Muestra por pantalla el contenido de todos los campos de la línea.

4. Mostrar datos de usuario.

(i)Pide el nombre y apellido del usuario cuya información se desea obtener, (2) Si existe algún usuario con ese nombre y apellido, muestra por pantalla la ciudad y el país asociados al mismo. Se mostrarán la información relativa a todos los usuarios, cuyo nombre y apellido coincidan con el introducido. Si no existe ningún usuario con esas características, se muestra un mensaje indicando esta eventualidad.

5. Salir del programa

La aplicación muestra el mensaje “Adiós.” y finaliza.

Requisitos:

- a) Los nombres de los ficheros deberán establecerse en la aplicación como una opción `#define` de preprocesado.
- b) El fichero de entrada a utilizar es `fich01.csv`.

ANEXO

Funciones a utilizar:

```
int LeerFicheroE (char NomFichero[], tUsuario **e, int *Usuarios);
int EscribirFichero(char NomFichero[], tUsuario *e, int
NumUsuarios);
int EliminarLinea(tUsuario **e, int NumLinea);
int MostrarDatosDeUsuario(tUsuario **e);
```

El alumno puede añadir otras funciones si las considera necesarias.

3.2 Programa 2

Se construirá un programa que lea **dos** vectores de números de longitud variable (longitud que deberá ser escogida por el usuario, requiriéndose asignar dinámicamente memoria a ambos vectores), seguidamente el programa calculará su producto escalar con otra función, que devuelve resultado al programa principal. El programa mostrará por pantalla tanto el resultado del producto escalar como los dos vectores.

El proceso descrito se repetirá un número indeterminado de veces según voluntad del usuario.

No olvidar liberar la memoria asignada dinámicamente cuando corresponda.

El producto escalar de dos vectores **vec1** y **vec2** de dimensión **n** es:

$$pe = \sum_{i=1}^n vec1[i] \cdot vec2[i]$$

Los prototipos de las funciones que se utilizarán son:

- Función para leer cualquiera de los vectores:
`void LeerVector (float *vector, int n);`
- Función para calcular el producto escalar de dos vectores **v1** y **v2**, y devuelve dicho valor:
`void ProductoEscalar (float *v1, float *v2, int n, float *ppe);`
- Función para mostrar cualquiera de los vectores:
`void ImprimirVector (float *v, int n);`

Queda prohibida la difusión de este material y de cualquier parte de su contenido fuera del ámbito de la UFV.