Deusto
Facultad de Ingeniería

EXAMEN PROGRAMACIÓN IV

(Parcial lenguaje C, 03 de abril de 2017)

Nombre y apellidos: _____

Código individual: ____

INSTRUCCIONES PARA EL EXAMEN

- La entrega del examen resuelto se realiza a través de la plataforma ALUD2. Existe una tarea para la entrega del examen llamada Entrega examen 03/04/2017
- El alumno debe crear un único archivo ZIP que contenga TODO el proyecto C y los ficheros requeridos, y subirlo a la plataforma. Solamente se corregirán los archivos entregados. Comprobar contenido del ZIP.
- Es imprescindible que en el fichero correspondiente al programa principal se incluya en la parte superior, como comentario, el nombre del alumno y el código individual que se entregará en el examen.
- La duración del examen es de 1 hora y 30 minutos.
- Se puede hacer uso de la referencia de la librería de C contenida en ALUD2
- Se permite el uso de cualquier material en el examen (prácticas y apuntes).

Deusto
Facultad de Ingeniería

Se parte de un módulo "persona" ya creado (con su fichero de cabecera "persona.h" e implementación "persona.c") que declara la estructura **`Persona`** (una cadena de caracteres para el nombre de la persona y un entero para su edad) y un fichero "main.c" que contiene el programa principal. Este programa principal crea e inicializa un array de cinco personas. Todo este código que ya se te proporciona NO debe ser alterado en ningún momento.

Antes de comenzar el examen, prueba que este mínimo programa funciona correctamente de modo que estés seguro que el entorno de desarrollo se encuentra bien configurado. Una vez hecha esta comprobación, el examen consta de tres partes que deberás ir resolviendo incrementalmente.

NOTA: los parámetros que se indican con la expresión `<TYPE>` a lo largo de este enunciado para las distintas funciones a implementar en el examen son orientativos, es decir, no se detalla en ningún caso si éstos deben ser punteros o estructuras. Será el alumno quien deba decidir y actuar en consecuencia. En muchos casos existirán varias alternativas correctas.

PARTE 1 (1,25 puntos)

1.1 Crea un nuevo módulo "censo" (con su correspondiente fichero de cabecera y de implementación), el cual defina la estructura **`GrupoPersonas`**. Esta estructura deberá contener el número de personas que componen el grupo, un array con las personas del grupo (deberás hacer uso de la estructura **`Persona`** que ya se proporciona) y su media de edad. Ten presente que el número de personas del grupo no es un dato conocido a priori.

1.2 Añade dentro del módulo "censo" estas tres funciones (el argumento `tamanyo` indica en todos los casos la longitud del array recibido):

> **`int cuantasPersonas(<array-personas> ap, int tamanyo, int edad)`**: calcula cuántas personas menores de la edad indicada como tercer parámetro hay en el array de personas. La función devuelve el número de personas encontradas.
>
> **`<GrupoPersonas> recuperarJovenes(<array-personas> ap, int tamanyo)`**: busca en el array de personas aquellas cuya edad sea menor que 30 años y crea y devuelve un grupo con estas personas y su media de edad.
>
> **`Persona* recuperarYogurin(<array-personas> ap, int tamanyo)`**: busca la persona de menos edad en el array y la devuelve. Asumimos que sólo hay una persona con la menor edad.

PARTE 2 (0,75 puntos)

2.1 Dentro de tu programa principal y usando el array de personas que se te proporciona, llama a las tres funciones implementadas en el módulo "censo" y visualiza sus resultados.

Juego de ensayo. Asumiendo en el primer caso una consulta del número de personas menores de 15 años, el resultado por pantalla sería el siguiente:

```
Personas menores de 15 años = 2
Grupo de personas menores de 30 años:
[Nombre: Hodei, Edad: 6]
[Nombre: Aitor, Edad: 12]
[Nombre: Maite, Edad: 24]
Media: 14.00
La persona más joven es:
[Nombre: Hodei, Edad: 6]
```

2.2 Implementa en el programa principal una función como la siguiente:

> **`void crearInforme(<GrupoPersonas> gp, char* fichero)`**: vuelca a un fichero de texto cuyo nombre se pasa como segundo parámetro los datos del grupo de personas pasados como primer parámetro.

2.3 En el programa principal usa la función anterior para generar un informe (fichero de texto) con los datos del grupo de personas jóvenes (menores de 30 años) que obtuviste en el apartado 2.1.

Juego de ensayo. En el caso anterior el contenido del fichero generado debería ser el siguiente:

```
CENSO DE JOVENES
----------------
[Nombre: Hodei, Edad: 6]
[Nombre: Aitor, Edad: 12]
[Nombre: Maite, Edad: 24]
Media: 14.00
```

Sólo en caso de que en el apartado 1.2 no hayas conseguido implementar correctamente la función `recuperarJovenes`, puedes crear e inicializar un grupo de personas con los valores que quieras y usarlo como argumento de la función `crearInforme` para cumplir con este apartado.

2.4 *Opcional (0,25 puntos adicionales)*. Hacer que en la primera consulta en la que se obtiene el número de personas menores de una determinada edad, en lugar de ser siempre 15 años y estar este valor incrustado en el código, se lea de los argumentos que recibe el programa.

Deusto
Facultad de Ingeniería

PARTE 3 (1 punto)

3.1 Añade dentro del módulo "censo" las siguientes funciones:

> **`int cuantosNombres(<array-personas> ap, int tamanyo, char letra)`**: cuenta cuántas personas hay en el array cuya inicial del nombre (consultar simplemente la primera letra) coincida con la letra recibida como parámetro y devuelve ese valor.
>
> **`char** listadoNombres(<array-personas> ap, int tamanyo, char letra)`**: encuentra en el array las personas cuya inicial del nombre coincida con la letra recibida como parámetro y devuelve el conjunto de los nombres de estas personas.

3.2 En el programa principal y usando el array de personas del que partías al principio del examen llama a las dos funciones anteriores para consultar los nombres que empiezan por la letra 'A'. Visualiza por pantalla los nombres encontrados.

Juego de ensayo. En el caso anterior el resultado por pantalla sería el siguiente:

```
Nombres con la inicial 'A':
Anita
Aitor
```

3.3 Finalmente, no olvides liberar la memoria de aquellas estructuras que así lo requieran, si las hubiera.