

Se incluyen dos versiones incompletas de un programa «progexamen»: una en C y otra en ensamblador MIPS. El programa, una vez completo, mostrará un menú donde se nos ofrecerán varias opciones para probar los ejercicios del examen que se describen a continuación.

Los códigos C y ensamblador que se proporcionan incluyen el menú y algún código auxiliar y de prueba, además de la cabecera de los procedimientos que se deberán implementar en ensamblador durante el examen. Sin embargo, no se incluye en la versión en C el código de algunos procedimientos que deberán ser diseñados e implementados durante el examen, por lo que no se puede compilar el programa dado. Por ello, también **se puede comprobar el funcionamiento que debe tener el programa una vez terminado en** <https://ditec.um.es/~rfernandez/etc-demo-examen/G32ME2/>.

Teniendo en cuenta lo anterior, haga lo que piden los siguientes apartados, implementando los procedimientos que se piden en el fichero progexamen.s:

1. (2 puntos) **Corrija** los errores de la traducción del procedimiento `ordenar_pacientes_por_nacimiento`, añadiendo un comentario al final del procedimiento explicando muy brevemente cada error corregido. Este procedimiento se llama desde la opción número 1 del menú y ordena el array `Pacientes`. Para que se considere correcto el ejercicio, el programa resultante debe respetar todos los convenios de programación vistos en clase y debe ejecutarse sin generar ningún aviso (*warning*) ni error. La traducción de todas las variables globales del segmento `.data` es correcta (es decir: todos los errores están en las instrucciones del programa, no en los datos).

```
void ordenar_pacientes_por_nacimiento(void);
```

2. (4 puntos) **Traduzca** el procedimiento `calcular_estadisticas`, cuyo código se proporciona en el programa en C. Este procedimiento calcula estadísticas anuales sobre pacientes y sus consultas, cuyos datos se encuentran almacenados en el array `consultas`. El procedimiento recibe dos argumentos: el año del que se desea obtener las estadísticas y un puntero a una estructura de tipo `Estadisticas` en la cual se almacenan los resultados.

```
void calcular_estadisticas(int ano, Estadisticas *estadisticas);
```

3. (4 puntos) **Implemente** el procedimiento `mostrar_y_contar_consultas`, que recibe como parámetro un paciente sobre el que se hace la consulta. Este procedimiento debe imprimir por pantalla la fecha y el informe de cada una de las consultas que hay almacenadas para dicho paciente. Para ello, consulta el array `consultas` buscando todos los registros correspondientes al paciente (es decir, aquellas consultas para las que `paciente_id` sea igual al campo `id` del paciente). Además, debe devolver un entero indicando el número de consultas encontradas para el paciente.

```
int mostrar_y_contar_consultas(Paciente* paciente);
```

El formato con el que se deben imprimir la fecha e informe de cada consulta es el siguiente:

```
yyyy-mm-dd informe
yyyy-mm-dd informe
...
```

Por ejemplo, para una consulta con fecha 5 de mayo de 2007 e informe «migrañas.» el texto impreso sería:

```
2007-10-05 migrañas.
```

Para la realización de este ejercicio, se pueden utilizar si se desea los procedimientos `mostrar_fecha`, `print_string`, y `print_character` ya incluidos en el programa.

Notas importantes a tener en cuenta para la realización de los ejercicios:

- Los procedimientos auxiliares y el menú están bien implementados y no es necesario ni recomendable gastar tiempo en entender cómo están implementados. Solo hay que usarlos para comprobar el funcionamiento de los procedimientos implementados durante el examen.
- Para poder realizar los ejercicios, debe **tener en cuenta las definiciones de los diferentes tipos de datos** utilizados que **se encuentran en la versión en C** del programa.
- En el caso de los ejercicios en los que se pide que se traduzca una función, la traducción debe ser lo más literal posible y se debe evitar cualquier tipo de optimización (en particular, debe respetar fielmente la estructura de los bucles originales).
- En el caso de los ejercicios en los que se pide que se implemente alguna función, se pueden utilizar funciones auxiliares siempre que se considere oportuno.
- A la hora de evaluar, solo se tendrá en cuenta el código ensamblador (se ignorará cualquier modificación de la versión en C). **Como respuesta al examen debe entregar solo el fichero progexamen.s modificado.**
- Puede comprobar si su solución se comporta como la solución correcta comparando la salida de su programa con la versión *online*, pero tenga en cuenta que el código de la solución puede ser incorrecto aun cuando se comporte correctamente (por ejemplo, si no se han seguido correctamente todos los convenios de programación o si se ha realizado una traducción incorrecta).