



## Estructura y Tecnología de Computadores

### Examen final de prácticas (3 de junio de 2021, turno segundo)

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/2021-06-turno2/>.

Teniendo en cuenta lo anterior, realice 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 `listar_facturas`, 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 muestra una lista de las facturas disponibles junto con información de cada una de ellas (esta información es el número de artículos que incluye la factura y el importe total). 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.

```
void listar_facturas(void);
```

2. (4 puntos) **Traduzca** el procedimiento `ordenar_articulos_por_valor_total`, cuyo código se proporciona en el programa en C. Este procedimiento no recibe ningún parámetro ni devuelve ningún valor, solo ordena los elementos del array `articulos` de forma descendente según el valor total de cada artículo. Este valor total se calcula multiplicando el número de unidades disponibles del artículo por su precio unitario. Este procedimiento se llama desde la opción número 2 del menú

```
void ordenar_articulos_por_valor_total(void);
```

3. (4 puntos) **Implemente** el procedimiento `actualiza_precio` que recibe la dirección de un array de estructuras de tipo `Articulo` (`articulos`), el número de elementos almacenados en el array (`num_articulos`), un porcentaje de 0 a 100 (`porcentaje`) y una cantidad máxima de producto (`cantidad_maxima`). El campo `unidades` de cada elemento del array `articulos` indica cuántas unidades de dicho artículo quedan en stock. El procedimiento debe incrementar en el porcentaje indicado el precio de aquellos artículos de los cuales quede una cantidad igual o inferior al parámetro `cantidad_maxima`.

Para calcular este incremento, utilice aritmética de enteros y asegúrese de que las multiplicaciones se hacen antes que las divisiones enteras. Por ejemplo, el 10 % (siendo 10 el valor del argumento de entrada `porcentaje`) de un precio `p` se debe calcular como  $(p10)/100$ .

```
void actualiza_precio(Articulo* articulos, int num_articulos, int porcentaje, int cantidad_maxima);
```

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 aún 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).