



Estructura y Tecnología de Computadores

Examen final de prácticas (29 de junio de 2021)

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/XVAHJEO/>.

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_ciudades_alfabeticamente`, 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 `ciudades` según el orden alfabético del campo `nombre_ciudad`. 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 ordenar_ciudades_alfabeticamente(void);
```

2. (4 puntos) **Traduzca** el procedimiento `seleccionar_ciudades_por_pais`, cuyo código se proporciona en el programa en C. Este procedimiento recibe dos argumentos: el nombre de un país (`pais`, que es un puntero a un array de caracteres), y la dirección de memoria de comienzo de un array en el que debe almacenar las ciudades que pertenezcan a ese país (`array_destino`). La lista de ciudades a seleccionar se encuentra inicialmente en el array global `ciudades`.

```
int seleccionar_ciudades_por_pais(char* pais, Ciudad* array_destino);
```

3. (4 puntos) **Implemente** el procedimiento `actualiza_poblacion` que recibe un número de años (`num_anos`) y actualiza la población de las ciudades del array `ciudades`. Para cada ciudad, cada año la población se incrementa en el porcentaje indicado por el campo `porcentaje_variacion_anual`. El procedimiento debe calcular la variación anual de la población de cada ciudad y sumárselo al valor actual, repitiendo el proceso tantas veces como indique `num_anos`. Además, debe acumular el valor calculado de cada variación para devolver al final de todo el proceso cuánto se ha incrementado la población total de las ciudades a lo largo de `num_anos` años.

```
int actualiza_poblacion(int num_anos);
```

Para calcular la variación anual de la población de una ciudad, utilice aritmética de enteros y asegúrese de que las multiplicaciones se hacen antes que las divisiones enteras. Por ejemplo, el 10% (siendo este valor del campo `porcentaje_variacion_anual`) de una población p (siendo p el campo `poblacion`) se debe calcular como $(p \times 10)/100$. Los redondeos se deben realizar por truncamiento (que es el comportamiento habitual de la aritmética entera).

Por ejemplo, dadas las siguientes ciudades:

Ciudad	País	Poblacion	Porcentaje de variación anual
Madrid	España	3 223 334	5
Murcia	España	447 182	-2

Al cabo de un año la variación de Madrid será de $(3\,223\,334 \times 5)/100 = 161\,166,7$, que es 161 166 una vez redondeado (este será el valor devuelto por la instrucción de división entera). La variación de Murcia será de $(447\,182 \times (-2))/100 = -8\,943,64$, que son -8 943 redondeado. Por tanto, el segundo año se parte de la siguiente situación:

Ciudad	País	Poblacion	Porcentaje de variación anual
Madrid	España	$3\,223\,334 + 161\,166 = 3\,384\,500$	5
Murcia	España	$447\,182 + (-8\,943) = 438\,239$	-2

El mismo proceso se debería repetir tantas veces como indique num_anos.

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 de 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).