



Estructura y Tecnología de Computadores

Examen parcial de prácticas (1 de marzo de 2023)

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/YY2JLC/>.

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

1. (5 puntos) **Traduzca** el procedimiento factorizar_potencia_2, cuyo código se proporciona en el programa en C. Este procedimiento recibe como parámetro un entero, n, y lo descompone en potencias de 2. El procedimiento no devuelve nada. Obsérvese que en progexamen.s no se ha reservado espacio para la variable maxima_potencia, por lo que **se deben añadir las directivas necesarias** para ello en el lugar adecuado.

```
void factorizar_potencia_2(int n);
```

2. (5 puntos) **Implemente** el procedimiento generar_serie, el cual recibe cuatro argumentos, x0, n, y y op, y no devuelve nada. Este procedimiento debe comprobar que op sea '+' o '*' e imprimir el mensaje "\nOperación inválida\n" en caso contrario. Si op es un carácter válido, el procedimiento debe imprimir los elementos $x_0..x_n$ tal que $x_n = x_{n-1}$ OP y.

```
void generar_serie(int x0, int n, int y, char op);
```

A continuación se muestran dos ejemplos del funcionamiento de este procedimiento:

```
Introduce x0: 2
Introduce n: 5
Introduce y: 3
Introduce la operación ('+' o '*'): +
Serie: 2 5 8 11 14 17
```

```
Introduce x0: 2
Introduce n: 3
Introduce y: 2
Introduce la operación ('+' o '*'): *
Serie: 2 4 8 16
```

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. Sólo hay que usarlos para comprobar el funcionamiento de los procedimientos implementados durante el examen.
- Para poder realizar los ejercicios, debe de **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 hecha en 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).