

Programación de Sistemas para Control de Procesos.

Ingeniería Técnica Industrial en Electrónica

NOMBRE: _____

1. Deseamos crear una estructura de datos dinámica de tipo **PILA**, en la que cada nodo almacena un número real. Efectuar lo siguiente (4p.):
 - Definir la estructura de datos e implementar sólo las dos siguientes operaciones primitivas para manejar ese tipo de dato abstracto: **Introducir** datos en la pila y **extraer** datos de la pila. Ambas primitivas deben devolver un código de error si éste se produce.
 - Hacer un programa principal que utilice el tipo de dato PILA, para efectuar una serie de operaciones. El programa leerá (por teclado) una instrucción (que será una cadena) y efectuará sobre la PILA la operación que dicha instrucción indique. Las operaciones sobre la PILA son las siguientes:
 - SCN: Lee un dato por teclado y lo introduce en la cima de la pila.
 - PTR: Muestra el dato situado en la cima de la pila.
 - ADD: Suma los 2 elementos situados en la cima de la pila (sacándolos de ella) e introduce el resultado en la pila.
 - SUB: Resta los 2 elementos situados en la cima de la pila e introduce el resultado en la pila
 - MUL: Multiplica los 2 elementos situados en la cima de la pila e introduce el resultado en la pila
 - DIV: Divide los 2 elementos situados en la cima de la pila e introduce el resultado en la pila
 - END: Termina la ejecución del programa.
 - El programa debe indicar cualquier error que se pueda producir y, en caso de producirse un error, no debe alterar el estado anterior de la pila. Por ejemplo, si hacemos un ADD y sólo hay un elemento en la pila, debe indicar que no se puede efectuar dicha operación y debe dejar la pila en el mismo estado en el que estaba.
2. Implementar una función en C, sin argumentos, que devuelva un valor de tipo `unsigned long`. La función devolverá, en cada llamada, sucesivamente un término de la sucesión de Fibonacci. Los dos primeros términos de la sucesión de Fibonacci son 1, y a partir de éstos, el siguiente término se obtiene sumando los dos anteriores, obteniendo: 1, 1, 2, 3, 5, 8, 13, 21, 34... (1.5 p.).
3. Defina una macro (instrucción del preprocesador de C), para simplificar el uso de la función `malloc()`. La macro se debe llamar `MALLOC` y debe tener 2 argumentos: El primero será un número `n` y el segundo un tipo. La macro deberá llamar a la función `malloc()` para reservar espacio suficiente para almacenar `n` valores del tipo indicado. El resultado de la macro debe ser un puntero al tipo indicado (usar un molde). Por ejemplo, si `x` es un puntero a entero, sería correcta la siguiente asignación (1.5 p.): `x=MALLOC(5, int);`
4. Indique a la izquierda de cada afirmación si es Verdadera (V) o Falsa (F), (3 p.):
(Correcta: 0.3p.; Incorrecta: -0.2p.; Blanca: 0p.)
 - ___ No siempre es necesario liberar la memoria asignada dinámicamente, ya que al terminar un programa siempre se libera automáticamente.
 - ___ La función para insertar en una lista dinámica es más simple de programar si la lista no tiene cabecera.
 - ___ El concepto de las estructuras de datos pila y cola sólo pueden ser implementados como estructuras de datos dinámicas y nunca como estructuras de datos estáticas.
 - ___ Un variable con el especificador de clase de almacenamiento `register` se almacena siempre en un registro del procesador.
 - ___ Los operadores a nivel de bits sólo pueden usarse con datos de tipos numéricos.
 - ___ Los miembros de un campo de bits (*bit fields*) sólo pueden ser de tipo `int`, `signed int` o `unsigned int`.
 - ___ Los ficheros con extensión `.h` son ficheros que incluyen las implementaciones de las funciones de una biblioteca (o librería).
 - ___ La siguiente instrucción escribe 1: `printf("%d", (5+4) && (3<1000));`
 - ___ La siguiente instrucción escribe 4: `printf("%d", 2<<1);`
 - ___ De la siguiente expresión `v[i].x` se puede deducir que `v` es un array de estructuras.