

TRABAJO PRÁCTICO N° 7

El TP7 es de práctica y no contiene ejercicios que deban ser entregados.

1. Escribir el código de la función recursiva `dec2bin()` que recibe un número como argumento y lo imprime en la pantalla en binario.
2. Escribir de manera recursiva una función que devuelve si un arreglo es capicúa (es decir, sus elementos están espejados). El prototipo debe ser:

```
int is_capicua(int* p1, int *p2);
```

Siendo p1 un puntero al primer elemento de un arreglo y p2 uno al último.

3. Analizar la siguiente función:

```
int suma_digitos(int n1)
{
    return ((n1 % 10) + digitos(n1 / 10)); //caso recursivo
}
```

- ¿Cuál creen que es el funcionamiento deseado de la función?
 - ¿Por qué no funciona? Corregirla.
4. Escribir una función recursiva que verifique si una cadena es un palíndromo (se lee igual al derecho y al revés). La función debe ignorar espacios y signos de puntuación. La función debe ser recursiva, es decir, debe estar estructurada con caso base y caso recursivo. Algunos ejemplos de palíndromos válidos:
 - "Anana"
 - "Anita lava la tina"
 - "La ruta nos aporoto otro paso natural"
 - "A man, a plan, a canal, Panama"
 5. Sea un contador módulo 4 (0-1-2-3) que cuenta hacia arriba si la entrada es 1 o hacia abajo si es 0. La salida es 1 si hay *overflow* o *underflow* y 0 si no.

Realizar el **diagrama de estados** de la FSM, indicando estado, entrada y salida.

6. A una FSM ingresa una secuencia binaria de 1 y 0. La salida cambia de estado tras cada flanco descendente (es decir, cuando ingresa un 0 inmediatamente luego de un 1).

Un ejemplo de su funcionamiento es:

IN	0	1	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	0
OUT	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1

Realizar el **diagrama de estados** de la FSM, indicando estado, entrada y salida.

7. Sea una FSM a la que ingresa de manera consecutiva una secuencia de 1 y 0. Dicha máquina filtra esta señal, sacando un 1 si al menos 2 de los últimos 3 valores recibidos fueron un 1.

Un ejemplo de su funcionamiento es:

IN	0	0	1	0	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0
OUT	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Realizar el **diagrama de estados** de la FSM, indicando estado, entrada y salida.

8. Diseñar una FSM que represente el funcionamiento de un reloj digital con dos botones:
- El botón 1 cambia el modo del reloj: de mostrar la hora actual, a ajuste de horas, a ajuste de minutos.
 - El botón 2 permite incrementar las horas o minutos en el estado correspondiente.

Realizar el **diagrama de estados** de la FSM, indicando estado, entrada y salida.

9. Diseñar una FSM que reciba en forma serial un archivo de código C. Se desea que la salida sea el mismo archivo pero con todos los comentarios suprimidos.

(Recordar que hay dos tipos de comentarios en C: /* multilínea */ y //una línea)

- Realizar el **diagrama de estados** de la FSM, indicando estado, entrada y salida.
- Implementar** la FSM utilizando código en C.