

Resolver los siguientes problemas en forma clara y legible en código ISO-C99.

1. Teniendo representado el TDA lista **de enteros** como `typedef struct {struct nodo *prim;} lista_t;` y el nodo `struct nodo {struct nodo *sig; int dato;};` implementar una primitiva

```
void lista_reducir(lista_t *l, int (*f)(int a, int b));
```

que para dos nodos de la lista los retire y aplique la función `f` a los mismos guardando en la lista el resultado. La función debe aplicarse reiteradas veces hasta que quede un único elemento en la lista.

Por ejemplo, si la lista es [1, 2, 3, 4] y `f` es `int f(int a, int b) {return a + b;}` la lista reducida será [10].

2. Un dispositivo de adquisición de datos se comunica mediante protocolo I2C. Para realizar la transmisión de un byte el mismo se envía en dos paquetes distintos, uno con su parte alta y otro con su parte baja. El paquete para la parte alta es 01000xxx donde xxx son los 3 bits más pesados del byte a transmitir, mientras que el paquete de la parte baja es 011xxxxx donde xxxxx son los 5 bits más livianos del byte a transmitir.

Por ejemplo si se quisiera transmitir el byte 10111001 se deberá codificar como los paquetes 01000101 y 01111001.

- a. Implementar la función `uint8_t obtener_byte(uint8_t alta, uint8_t baja);` que reciba la parte alta y la parte baja de una transmisión y devuelva el byte recibido.
  - b. Implementar la función `void generar_byte(uint8_t byte, uint8_t *alta, uint8_t *baja);` que reciba un byte a transmitir y devuelva la parte alta y baja de los paquetes a enviar.
3. Escribir una función `bool esta_ordenado(float v[], size_t n);` que utilizando recursividad indique si un vector `v` de `n` elementos se encuentra ordenado o no.

¡Suerte! :)

Resolver los siguientes problemas en forma clara y legible en código ISO-C99.

1. Teniendo representado el TDA lista **de enteros** como `typedef struct {struct nodo *prim;} lista_t;` y el nodo `struct nodo {struct nodo *sig; int dato;};` implementar una primitiva

```
void lista_reducir(lista_t *l, int (*f)(int a, int b));
```

que para dos nodos de la lista los retire y aplique la función `f` a los mismos guardando en la lista el resultado. La función debe aplicarse reiteradas veces hasta que quede un único elemento en la lista.

Por ejemplo, si la lista es [1, 2, 3, 4] y `f` es `int f(int a, int b) {return a + b;}` la lista reducida será [10].

2. Un dispositivo de adquisición de datos se comunica mediante protocolo I2C. Para realizar la transmisión de un byte el mismo se envía en dos paquetes distintos, uno con su parte alta y otro con su parte baja. El paquete para la parte alta es 01000xxx donde xxx son los 3 bits más pesados del byte a transmitir, mientras que el paquete de la parte baja es 011xxxxx donde xxxxx son los 5 bits más livianos del byte a transmitir.

Por ejemplo si se quisiera transmitir el byte 10111001 se deberá codificar como los paquetes 01000101 y 01111001.

- a. Implementar la función `uint8_t obtener_byte(uint8_t alta, uint8_t baja);` que reciba la parte alta y la parte baja de una transmisión y devuelva el byte recibido.
  - b. Implementar la función `void generar_byte(uint8_t byte, uint8_t *alta, uint8_t *baja);` que reciba un byte a transmitir y devuelva la parte alta y baja de los paquetes a enviar.
3. Escribir una función `bool esta_ordenado(float v[], size_t n);` que utilizando recursividad indique si un vector `v` de `n` elementos se encuentra ordenado o no.

¡Suerte! :)