

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

1. Teniendo representado el TDA lista **genérica** como `typedef struct {struct nodo *prim;} lista_t`; y el nodo `struct nodo {struct nodo *sig; void *dato;}`; implementar una función

`lista_t *lista_filtrar(lista_t *l, bool (*f)(void *));`

que retorne una nueva lista que contenga todos los elementos de `l` donde la función `f` aplicada al dato devuelva `true`. Los elementos deben ser retirados de la lista `l`.
2. En las elecciones del Centro de Estudiantes se abren urnas en Paseo Colón y en Las Heras con el mismo padrón. Se tiene un vector con todos los padrones que votaron en PC y otro con todos los padrones que votaron en LH ambos ordenados por número de padrón creciente. Escribir una función:

`int *detectar_duplicados(const int pc[], size_t n_pc, const int lh[], size_t n_lh, size_t *n_dup);`

que, de forma eficiente, calcule y devuelva un vector con los padrones que duplicaron su voto en ambas sedes. Debe devolver en `n_dup` el tamaño del vector resultante. (**Nota:** No importa si sobra memoria en el vector.)
3. Implementar una función `float sumar_elementos(float v[], size_t n)`; que utilizando recursividad calcule la suma de los elementos de un vector `v` de `n` flotantes.

¡Suerte! :)