

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 función

```
void unir_listas_ordenadas(lista_t *d, lista_t *o);
```

que reciba dos listas ordenadas `d` y `o` y que junte ambas listas en `d`. La lista resultante debe quedar ordenada y la lista `o` debe quedar vacía.

(**Nota:** Sólo se puede recorrer las listas una única vez.)

2. Se quiere implementar un TDA que sea capaz de almacenar 50 valores booleanos indexados por un entero entre 0 y 49.

Se propone implementar el TDA sobre la declaración `typedef uint64_t booleanos_t;` y utilizar los bits de este entero para guardar los estados de los diferentes booleanos.

- a. Implementar la primitiva `void booleanos_guardar(booleanos_t *b, int indice, bool valor);` que almacene el `valor` en la posición `indice`.
 - b. Implementar la primitiva `bool booleanos_obtener(const booleanos_t *b, int indice);` que devuelve el valor almacenado en la posición `indice`.
3. Implementar una función recursiva que responda al siguiente prototipo

```
bool es_palindromo(const char *s);
```

que reciba una cadena de caracteres `s` y que indique si la misma es palíndroma o no. Una cadena es palíndroma si se lee igual desde ambos lados, como por ejemplo "neuquen".

¡Suerte! :)