

Algoritmos y Programación I (95.11) – Curso Essaya – 4^{to} parcialito – 01/07/2022

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

1. Teniendo un TDA lista enlazada genérica definido como

```
typedef struct { struct nodo *prim; } lista_t;
```

y su nodo como

```
struct nodo { struct nodo *sig; void *dato; };
```

implementar una primitiva

```
bool lista_esta_ordenada(const lista_t *l, int (*cmp)(const void *, const void *));
```

que indique si la lista está ordenada.

2. En un país hipotético el gobierno se dio cuenta de que puede redefinir de forma automática sus políticas de subsidios a los servicios cruzando listados de diferentes organismos. Se tiene un listado ordenado por número de DNI de todos los titulares de servicios y se tiene otro listado, también ordenado por DNI, que tiene a todas las personas que cobran más de \$350.000 al mes. Implementar una función:

```
int *mantienen_subsidio(int titulares[], size_t n_titulares, int salarios_altos[],  
                        size_t n_salarios_altos, size_t *n_mantienen);
```

que reciba un vector con los `n_titulares` titulares de servicios, otro vector con los `n_salarios_altos` salarios_altos personas de altos ingreso, compute **de forma eficiente** y luego devuelva un vector con los `n_mantienen` que mantienen su subsidio.

3. Se desea saber si una cadena contiene los caracteres de otra cadena patrón de forma ordenada (pero no necesariamente consecutiva). Implementar una función:

```
bool cadena_contiene(const char *cadena, const char *patron);
```

que **usando recursividad** indique si la cadena `patron` se encuentra intercalada en la cadena `cadena`.

Por ejemplo: `cadena_contiene("anticonstitucional", "tonto") -> true`, `cadena_contiene("hola", "ola") -> true`, `cadena_contiene("ballena", "llenaba") -> false`.

¡Suerte! :)