

## Algoritmos y Programación I (95.11) – Curso Essaya – 4<sup>to</sup> parcialito – 30/07/2021

Resolver los siguientes problemas en un único archivo de código ISO-C99.

1. Teniendo un TDA lista enlazada ordenada de enteros definido como

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

y su nodo como

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

implementar una primitiva

```
bool lista_ordenada_completar(lista_ordenada_t *l);
```

que reciba una lista *l* y complete tanto los números faltantes como elimine los repetidos. Por ejemplo, si la función recibe la lista [1, 3, 3, 4, 6] la misma debe quedar como [1, 2, 3, 4, 5, 6].

En caso de una falla de memoria interrumpir el procesamiento y devolver **false**, no hace falta restaurar a *l* a su condición previa.

2. Implementar una función

```
void filtrar(void *v[], size_t *n, bool (*evaluar)(void *),  
             void (*eliminar)(void *));
```

que reciba un vector *v* de *\*n* elementos y que elimine del vector todos los elementos sobre los cuales la función **evaluar** devuelva **true**, cada elemento se destruye con **eliminar**. Al finalizar la operación el vector debe contener sólo los elementos no eliminados y debe actualizarse la nueva longitud.

3. Implementar una función **recursiva**

```
size_t buscar(const int v[], size_t n, int elem);
```

que dado un vector *v* de longitud *n* devuelva la posición del vector en la cual se encuentra el elemento *elem*. Si *elem* no se encontrara en *v* se debe devolver *n*.

Si bien cada uno de los ejercicios es independiente de los demás se deberán codificar todas las funciones en un único archivo de código fuente que será el que se entregue.

El examen es de elaboración personal, todo el código entregado debe ser realizado por el alumno.

El examen se envía a través del sistema de entregas de trabajos prácticos.

¡Suerte! :)