Estructuras de Datos y Algoritmos

Segundo Semestre 2020 - Primer Parcial - Sede Buceo - Matutino/Vespertino

- ### Ejercicio 1

a) Se pide definir el *tipo Lista Circular de Enteros (sólo el tipo, no las operaciones).*

b) Con el tipo definido en ***a)*** y accediendo a la representación interna implemente una función que dada una *Lista Circular de Enteros*, elimine todas las ocurrencias de un entero pasado como parámetro.

- ### Ejercicio 2

Dado el siguiente *TAD Lista de Enteros:*

```
typedef struct NodoLista * lista;
struct NodoLista{
        int dato;
        lista sig;
}

lista Null ();
// Crea la lista vacia.

lista Cons (lista l, int e);
// Inserta el elemento e al principio de la lista l.

bool IsEmpty (lista l);
// Retorna "true" si l es vacía, "false" en caso contrario.

int Head (lista l);
// Retorna si l no es vacía el primer elemento de la lista.

lista Tail (lista l);
// Retorna si la l no es vacía, la lista sin su primer elemento.
```

Implemente ***recursivamente*** las siguientes funciones utilizando exclusivamente las operaciones anteriores (*sin acceder a la representación interna*)

a)

```
lista Take_from_to (int i, int f, lista l);
// Retorna la lista resultado de tomar los elementos desde el i hasta el f.
// l no comparte memoria con la lista resultado.
// Se considera i = 1 al primer elemento.
// Pre: i <= f.
```

b)

```
int Min(lista l);
// Retorna el mínimo elemento de l.
// Pre: l no es vacía.
```

- ***Ejercicio 3***

Las llamadas *Listas Generales de Enteros* son listas encadenadas simples donde en cada elemento hay una *Lista Simple de Enteros* (*como la del Ejercicio 2*).

a) Defina el *TAD Lista General de Enteros*, incluyendo el tipo y sus operaciones *constructoras*, *selectoras* y de *predicado*.

b) Sin acceder a la representación interna *(usando las funciones del TAD definido en* ***a)*** *y las del TAD Lista Simple definido en el* ***Ejercicio 2****)*, implemente una función que dada una *Lista General de Enteros* y un *entero* pasado como parámetro, retorna *true* si este pertenece a alguna de las listas simples, *false* en caso contrario.

- ***Ejercicio 4***

a) Indicar justificando si en el siguiente ejemplo el algoritmo de *sorting* utilizado es estable.

Orden Original:

Nombre	***Subject***	***Fecha Enviado***
Juan López	Saludos….	2020-10-26 11:44
Ana Gómez	Re: Reunión 2/10	2020-10-26 12:37
Juan López	Re: Saludos….	2020-10-26 14:57
Juan López	Fwd: Imprimir	2020-10-27 07:31
Ana Gómez	Informe Mensual	2020-10-27 17:12
Juan López	Recordatorio	2020-10-28 21:02

Luego de ordenar por nombre:

Nombre	***Subject***	***Fecha Enviado***
Ana Gómez	Re: Reunión 2/10	2020-10-26 12:37
Ana Gómez	Informe Mensual	2020-10-27 17:12
Juan López	Saludos….	2020-10-26 11:44
Juan López	Fwd: Imprimir	2020-10-27 07:31
Juan López	Re: Saludos….	2020-10-26 14:57
Juan López	Recordatorio	2020-10-28 21:02

b) Describa una técnica de ordenación interna estable (*nombre, cómo funciona, pseudocódigo, tiempos de ejecución, ventajas y desventajas que tiene*) para ordenar un arreglo de enteros.