

Tecnólogo Informático – San José

Estructuras de Datos y Algoritmos

Primer Parcial 2021

- Completar *TODAS* las hojas con el nombre y el número de cédula. Numerarlas y escribir el total en la primer hoja.
- Escriba las hojas de un solo lado.
- No se puede utilizar material de ningún tipo. *Apagar celulares.*
- Sólo se contestan dudas acerca de la letra de los ejercicios.
- El parcial dura 2 horas.

Ejercicio 1 [30 puntos]

Considere las siguientes definiciones:

```
struct nodo_lista {                struct nodo_ocurrencias {
    int info;                      int info;
    nodo_lista * sig;              unsigned cant;
};                                  nodo_ocurrencias * sig;
                                   };
typedef nodo_lista * Lista;        typedef nodo_ocurrencias * ListaOcurrencias;
```

Se pide implementar iterativamente las siguientes operaciones:

Parte a) [15 puntos]

```
// Recorre la lista L, crea una lista con múltiples ocurrencias y la retorna
// Pre-Condición: L está ordenada de menor a mayor, pudiendo contener elementos repetidos
// Post-Condición: la lista resultante deberá estar ordenada de menor a mayor
ListaOcurrencias crear_lista_ocurrencias (Lista L);
```

Parte b) [15 puntos]

```
// Elimina a lo sumo 'cant' ocurrencias del nodo con información 'elem'
// Si 'cant' es mayor o igual a la cantidad de ocurrencias de 'elem' dentro de la lista
// entonces el nodo debe ser eliminado
void eliminar_ocurrencias (ListaOcurrencias L, int elem, unsigned cant);
```

Nota: asuma que para ambas partes las listas pasadas utilizan Celda Dummy. En la primera operación se deberá retornar una lista de múltiples ocurrencias con Celda Dummy.

Ejercicio 2 [10 puntos]

Dada la representación usual de listas se pide implementar recursivamente la siguiente operación:

```
// Retorna true si L contiene a todos sus
// elementos contiguos separados por 2 unidades
// Pre-Condición: L está ordenada de menor a mayor
bool separados (Lista L);
```

Ejemplos:

| | | |
|---------------|---|---------------|
| L = [1,3,6,8] | → | retorna true |
| L = [1,3,7,8] | → | retorna false |
| L = [3] | → | retorna true |