

PRÁCTICA DE ESTRUCTURAS MULTITENLAZADAS

1. Dada una lista doblemente enlazada, implemente un algoritmo que retorne la lista invertida
2. Dadas las listas lineales doblemente enlazadas L1 y L2 que se encuentran ordenadas ascendentemente, elabore un algoritmo que mezcle ambas produciendo L3 en el mismo orden.

```

Type Node* List
Type Register Node
Integer info
Node* pNext, pPrev
end

```

3. Implemente una función que reciba una lista circular doblemente enlazada por parámetro y un entero n y retorne el resultado de sumar el elemento anterior de la lista al elemento actual n veces de la siguiente manera:  
Dada L = {1, 4, 3, 2, 3} y n=2 Iteración 1: {4, 5, 8, 10, 13} Iteración 2: {17, 22, 30, 40, 53}  
Retorna L= {17, 22, 30, 40, 53}
4. Implemente una función que imprima los elementos de una matriz esparcida.
5. Dada una matriz esparcida M, elabore un algoritmo que imprima los elementos de la diagonal principal si estos están presentes.
6. Dada una matriz esparcida M y un entero n, elabore un algoritmo que realice la búsqueda de n e imprima las posiciones dónde se encuentra dicho valor.
7. Se encuentran n ejecutivos sentados en una mesa redonda. El jefe de la mesa requiera que todos tengan en sus manos un documento importante, pero solo hay uno. No todos los ejecutivos tienen la misma retentiva y por ende no pueden recordar los detalles del documento si no lo leen bien. Se desea que, dada una lista circular donde cada nodo "i" posee un entero k con la retentiva del i-esimo ejecutivo, implemente una función que permita saber cuantas vueltas a la mesa debe dar el documento para que todos los ejecutivos sepan todos los detalles.
8. Considere la siguiente implementación de una matriz esparcida:

```

clase Sparse
NodeCol* pColumns
end
Type Register NodeCol
Integer iColumn
Node* pFirst
end
Type Register Node
Integer Row
Real Info
Node* pNextCol
end

```

Si se conoce que las matrices a almacenar son de 120x120, con un máximo de 200 elementos no nulos, haga un análisis respecto a la conveniencia de usar la representación dinámica propuesta o una representación estática, para estas condiciones. ¿Cuándo es recomendable usar la representación estática?

9. Considere la siguiente definición para matrices esparcidas:

```

Type Register Node
Integer iColumna
Real rInfo
Node* pNextRow, pPrevRow
end
class Sparse
Array Elem of Node* [1 .. N]
end

```

- a) Analice en qué situaciones conviene usar este tipo de representación y en qué situaciones conviene usar una matriz estática.
  - b) Si se conoce que la matriz tiene 100 filas y 50 columnas, y en la matriz existirán cuando mucho 150 elementos no nulos ¿Cuál es la representación más eficiente en cuanto a uso de memoria?
  - c) ¿Bajo qué condiciones es más conveniente utilizar un arreglo bidimensional estático que una matriz esparcida dinámica? Justifique su respuesta basándose en los principios de complejidad en tiempo y complejidad en espacio.
10. La compañía DigitelUCV clasifica a sus clientes en A1, A2, A3, A4 y cada cliente puede contratar los servicios que desee: DigitelUCV básica, central privada, DigiAmigo, Digix, circuito privado.  
Se desea almacenar la información tal que se pueda conocer:

- a) ¿Cuántos clientes de un tipo determinado han contratado un servicio dado? De O(1)
- b) ¿Quiénes son esos clientes? De O(1)

Proponga una estructura de datos eficiente para satisfacer a) y b). Tome en cuenta que un cliente puede contratar más de un servicio a la vez, con clasificaciones distintas.

10. Diseñe un algoritmo que, dado el esquema de prelación de las materias de la carrera Computación y el código de una materia, imprima un listado con todas las asignaturas que prelan a la materia dada, en orden cronológico.