

Algoritmos y Estructuras de Datos.

3er Parcial. Tema: **1A.** [8 de julio de 2003]

[Ej. 1] [Primitivas (20 puntos)] Escribir las funciones del TAD DICCIONARIO listadas a continuación, implementado por tabla de dispersión cerrada con resolución lineal de colisiones ANULA(D), INSERTA(x,D), MIEMBRO(x,D) y SUPRIME(x,D). **Escribir todos los tipos, definiciones, funciones y procedimientos auxiliares necesarios.**

[Ej. 2] [Programación (total = 60 puntos)]

- (a) [MergeSort-Listas (30 puntos)] Escribir un procedimiento `procedure MERGESORT(var L:lista);` que ordena los elementos de una lista usando el método de clasificación por intercalación. Es decir, divide la lista en dos sublistas auxiliares L1 y L2 de tamaño similar, las ordena aplicando MERGESORT recursivamente y luego intercala las sublistas:
- dividir L en dos sublistas L1, L2
 - clasificar L1 y L2 por MERGESORT
 - intercalar L1 y L2 en L

Usar las primitivas del TAD LISTA: RECUPERA (p,L), SIGUIENTE (p,L), PRIMERO (L), y FIN (L) que considere necesarias.

- (b) [Selección para colas (30 puntos)] Escribir un procedimiento `procedure SELECCION (var C:cola);` que ordena los elementos de una cola, usando una cola auxiliar CAUX. El procedimiento va extrayendo sucesivamente el mínimo de los valores de la cola C y los va insertando en la cola CAUX hasta que C este vacía. Finalmente pasa todo el contenido de CAUX a C. Para extraer el mínimo de una cola escribir una función auxiliar `function SUPRIME_MIN (var C:cola): integer;` que retorna el mínimo de los elementos de una cola, suprimiéndolo, en donde se usa otra cola auxiliar C2. Usar las primitivas del TAD COLA: ANULA (C), PONE_EN_COLA (x,C), QUITA_DE_COLA (C), VACIA(C), y FRENTE_DE_COLA (C) que considere necesarias.

[Ej. 3] [Operativos (total = 20 puntos)]

- (a) [Heap-sort (10 ptos)] Dados los enteros {1, 5, 8, 2, 3, 0, 10, 4} ordenarlos por el método de “montículos” (“heap-sort”). Mostrar el montículo (minimal) antes y después de **cada** inserción/supresión.
- (b) [Bin-sort (10 ptos)] Dados los enteros {520, 811, 405, 71, 685, 23, 442, 536} ordenarlos por el método de urnas (“bin-sort”), con $B = 10$ urnas, y tantas pasadas como sea necesario. Mostrar la tabla de urnas y su contenido después de cada pasada. Especificar cuál es la instrucción Pascal que se puede utilizar para ubicar los elementos en las urnas durante cada pasada.