Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos. Examen Final. [7 de Agosto de 2003]

Ej. 1.- [**Primitivas (20 puntos)**] Escribir las funciones del `TAD ARBOL ORDENADO ORIENTADO` con celdas enlazadas por **punteros ó cursores** a saber: `PADRE(n,A)`, `HIJO_MAS_IZQ(n,A)`, `HERM_DER(n,A)`, `ETIQUETA(n,A)`, `CREA2(v,A1,A2)` y `ANULA(A)`. *Escribir todos los tipos, definiciones, funciones y procedimientos auxiliares necesarios.*

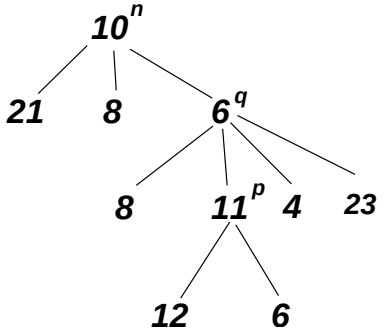
Ej. 2.- [**Ejercicios de programación (total 80 puntos)**]

(a) [**Encuentra (35 puntos)**] Escribir una función
`function ENCUENTRA(L1,L2: lista; var INDX:lista): boolean` que,

- Retorna `true` o `false` dependiendo de si `L1` es una sublista o no de `L2`.
- En caso de que si lo sea, retorna en `INDX` los índices de los elementos de `L2` que forman `L1`, si no `INDX` debe retornar vacía, independientemente de lo que contenía previamente.

Por ejemplo, si `L2`=$\{13, 9, 8, 12, 9, 6, 12, 2, 9, 14, 18, 10\}$ y `L1`=$\{13, 9, 9, 6, 2, 14\}$ entonces `ENCUENTRA` debe retornar `true`, y `INDX`=$\{1, 2, 5, 6, 8, 10\}$. Si `L1`=$\{8, 9, 13\}$ debe retornar `false` e `INDX`=$\{\}$. Nota: Los índices en `INDX` deben ser estrictamente crecientes. Utilizar las primitivas del TAD LISTA: `INSERTA(x,p,L)`, `RECUPERA(p,L)`, `SUPRIME(p,L)`, `SIGUIENTE(p,L)`, `ANULA(L)`, `PRIMERO(L)`, y `FIN(L)`.

(b) [**Contiene hijos (35 puntos)**] Escribir una función
`function CONT_HIJOS(L:lista; n:nodo; A:arbol): nodo` que retorna el nodo `m` del subárbol de `n` tal que las etiquetas de sus hijos corresponden exactamente con los enteros en la lista L. Si ningún nodo cumple esta condición entonces debe retornar Λ. Por ejemplo, en el árbol de la figura si `L`=$\{12, 6\}$ entonces `CONT_HIJOS(L,n,A)` debe retornar `p` mientras que si `L`=$\{8, 11, 4, 23\}$ debe retornar `q`. Por otra parte si `L`=$\{21, 8\}$ entonces debe retornar Λ ya que si bien `n` tiene como hijos a $\{21, 8\}$ también tiene al 6, el cual no esta en `L`. Usar las funciones del `TAD ARBOL ORDENADO ORIENTADO`: `HIJO_MAS_IZQ(n,A)`, `HERMANO_DER(n,A)`, `ETIQUETA(n,A)` y las del `TAD LISTA` `INSERTA(x,p,L)`, `RECUPERA(p,L)`, `SUPRIME(p,L)`, `SIGUIENTE(p,L)`, `ANULA(L)`, `PRIMERO(L)`, y `FIN(L)`.



(c) [**Invierte cola (10 puntos)**] Escribir un procedimiento
`procedure INVIERTE(var C:cola);` que invierte los elementos de una cola, usando una pila auxiliar. Por ejemplo, si `C`=$\{12, 4, 23, 10, 1\}$ entonces `INVIERTE(C)` debe dejar `C`=$\{1, 10, 23, 4, 12\}$. Utilizar las funciones del `TAD COLA`: `ANULA (C)`, `PONE_EN_COLA(x,C)`,

`QUITA_DE_COLA (C)`, `VACIA(C)`, `FRENTE_DE_COLA(C)` y del `TAD PILA ANULA(P)`, `METE(x,P)`, `SACA(P)`, `TOPE(P)` y `VACIA(P)`.

Ej. 3.- [**LIBRES. Ejercicios operativos (total 80 puntos)**] **Atención!! Alumnos libres deben completar un mínimo de 70% en cada uno de los ítems**

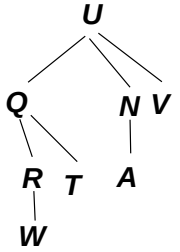
(a) [**Reconstruir árbol (25 puntos)**] Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son

- `ORD_PRE` $=\{Q, P, L, C, M, D, F, G, H\}$,
- `ORD_POST` $=\{P, C, M, F, G, H, D, L, Q\}$.

(b) [**Árboles de Huffman (20 puntos)**] Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario y encodar la palabra `PINGUINO`
$P(P) = 0.1, P(I) = 0.1, P(N) = 0.1, P(G) = 0.1, P(U) = 0.1, P(O) = 0.25, P(Z) = 0.25$
Calcular la longitud promedio del código obtenido.

(c) [**Heap-sort (25 puntos)**] Dados los enteros $\{3, 5, 2, 1, 6, 8, 3, 7\}$ ordenarlos por el método de *"montículos"* (*"heap-sort"*). Mostrar el montículo (minimal) antes y después de **cada** inserción/supresión.

(d) [**Particionar árbol (10 puntos)**] Considerando el árbol de la figura, decir cuales son los nodos `DESCENDIENTES(Q)`, `ANTECESORES(Q)`, `IZQUIERDA(Q)` y `DERECHA(Q)`.



Ej. 4.- [**LIBRES(20 ptos, 5 por pregunta)**] Responder según el sistema *"multiple choice"*, es decir marcar con una cruz el casillero apropiado. **Atención:** Algunas respuestas son intencionalmente "descabelladas" y tienen puntajes **negativos**!!

(a) ¿Cuál es el tiempo de ejecución para la función `ANTERIOR` para listas *doblemente enlazadas* en el peor caso? (n es el número de elementos en la lista)

☐ ... $O(1)$ ☐ ... $O(n)$ ☐ ... $O(n^2)$ ☐ ... $O(\log n)$

(b) ¿Cuál es el número de niveles en un árbol binario lleno (todos sus niveles están completos) en función del número n de nodos en el árbol?

☐ ... $O(1)$ ☐ ... $O(n)$ ☐ ... $O(n^2)$ ☐ ... $O(\log n)$

(c) ¿Cuál es el número de *intercambios* en el método de clasificación por seleccón?

☐ ... $O(1)$ ☐ ... $O(n)$ ☐ ... $O(n^2)$ ☐ ... $O(\log n)$

(d) ¿Cuál es el tiempo de ejecución del algoritmo de clasificación por montículos *en el peor caso*?

☐ ... $O(\log n)$ ☐ ... $O(n)$ ☐ ... $O(n^2)$ ☐ ... $O(n \log n)$