



6. Dado un árbol 'k-ario' (con un máximo de 'k' hijos por nodo) representado mediante punteros, ¿qué cantidad de enlaces estarán a **NIL** suponiendo 'n' número de nodos del árbol?

- a.- ' $n * (k-1) + 1$ '
- b.- ' $k * (k-1)$ '
- c.- ' $n * (k+1) - 1$ '
- d.- ' $2 * n + k$ '

7. Supongamos un montículo de mínimos de caracteres, en el que hemos insertado, por este orden y a partir de un montículo vacío, los siguientes caracteres: D, F, A, C, C, E (suponemos el criterio habitual de ordenación de caracteres, es decir,  $A < B < C < \dots < Y < Z$ .) ¿Cuál será el estado del montículo tras las siguientes operaciones:

Tras la inserción de los caracteres ('D', 'F', 'A', 'C', 'C', 'E')

H. Num:  H. Info:

Insertar ('G')

H. Num:  H. Info:

Insertar ('B')

H. Num:  H. Info:

Eliminar\_Mínimo

H. Num:  H. Info:

Eliminar\_Mínimo

H. Num:  H. Info:

8. Supongamos el grafo cuya matriz de adyacencia es:

	A	B	C	D	E	F
A	0	1	0	0	1	0
B	1	0	0	1	1	1
C	0	0	0	1	0	0
D	0	1	1	0	0	0
E	1	1	0	0	0	1
F	0	1	0	0	1	0

Indica el orden en que se recorrerán los nodos utilizando el recorrido BFS (en anchura) a partir del nodo 'A'.

9. Supongamos el tipo abstracto de datos (T.A.D.) tabla de valores, con las siguientes operaciones definidas:

CREAR\_TABLA ( ) → Tabla

ALMACENAR ( Tabla , Indice , Valor ) → Tabla

RECUPERAR ( Tabla , Indice ) → Valor

VALOR\_DEFINIDO ( Tabla , Indice ) → Boolean

¿Cuál de las siguientes afirmaciones es cierta?

- a.- Cualquier T.A.D. es independiente de su implementación, por lo que cualquier T.A.D. se puede representar mediante *arrays* y mediante punteros, excepto las tablas que son básicamente *arrays*.
- b.- El T.A.D. tabla, sólo puede ser implementado con punteros.
- c.- Cualquier T.A.D. es independiente de su implementación, por lo que cualquier T.A.D., incluso las tablas se puede representar mediante *arrays* y mediante punteros.
- d.- Una tabla, definida con estas operaciones y estos axiomas no puede ser considerado un tipo abstracto de datos.

Apellidos: Cognoms:	Nombre: Nom:
------------------------	-----------------

**Tiempo: 1 hora**      **Recomendación: Leer atentamente todo el examen antes de empezar a contestar**

Temps: 1 hora      *Recomanació: Llegir atentament tot l'examen avanç de començar a respondre*

No se aceptarán contestaciones escritas a lápiz, ni hojas de examen sin nombre      **No se permiten ni libros ni apuntes.**  
*No s'acceptarà cap contestació escrita en llapis , ni cap fulla d'examen sense nom*      No es permeten ni llibres ni apunts.

La puntuación para preguntas con múltiples opciones es: A

- Pregunta correcta: 1 punto
- Pregunta incorrecta: -0,2 puntos
- Pregunta en blanco: 0 puntos

Las preguntas en las que se marquen varias opciones serán consideradas incorrectas.

El examen se responderá en la misma hoja en el espacio reservado para ello.

**Cualquier respuesta fuera de estas hojas será ignorada.**

**En las preguntas con varias opciones marca la opción verdadera**

1. Enumera y justifica las desventajas de la utilización de un vector para la implementación de árboles binarios.

2. ¿Cuál es el coste espacial de una cola que contiene 20 elementos de tipo entero, suponiendo 2 bytes para los enteros, 4 para los punteros y 2 para los índices,...

a.- ...si suponemos una representación estática (mediante un <i>array</i> de 20 elementos)	b.- ...si suponemos una representación dinámica (mediante punteros.)
--	--

3. ¿Se puede hilvanar un árbol binario que no sea de búsqueda? Justifica la respuesta.

4. Una cola puede ser representada mediante una lista simplemente enlazada circular (sin nodo cabeza) ¿Es posible utilizar un solo puntero para trabajar con la cola (en vez de los dos habituales al inicio y al final de la cola)?. Si la respuesta es negativa explica el por qué. Si la respuesta es afirmativa ¿A qué elemento debería apuntar ese único puntero? ¿Por qué?.

5. Queremos obtener un listado de toda la información contenida tanto en un árbol binario de búsqueda como en una lista simplemente enlazada ordenada.

- a.- El coste temporal de la operación obtener listado es menor en el árbol binario de búsqueda, ya que en general, las operaciones en el árbol binario de búsqueda son del orden de ***log n***.
- b.- El coste temporal de la operación obtener listado es menor en la lista ligada ordenada.
- c.- El coste temporal de la operación obtener listado es el mismo en los dos casos.
- d.- Es imposible obtener el listado de toda la información contenida en el árbol binario de búsqueda, ya que los procedimientos definidos sobre el árbol sólo incluyen la búsqueda, la inserción y la eliminación de elementos.

**1. (5 punto)**

Realizar una función o procedimiento en Pascal que nos determine la media de los valores almacenados en los nodos interiores (no son hojas) un árbol binario. El prototipo de la función será:

```
Function Media ( arb: Arbol ): Real;
```

Suponemos que el tipo Valor guardado en el árbol son enteros. Escribe el código de aquellas funciones auxiliares que creas conveniente utilizar, así como la declaración de tipos del árbol.

**2. (5 punto)**

Supongamos el tipo Grafo definido de la siguiente manera:

```
Const
    MAX_NODOS = 100;

Type
    Indice = 1..MAX_NODOS

    Valor_Nodo = Char
    Valor_Arco = Char

    Punt_Arco = ^Arco

    Arco = Record
        Info      : Valor_Arco;    { Información asociada a cada arco }
        Destino   : Indice;
        Sig_Arco : Punt_Arco;
    End;

    Nodo = Record
        Info      : Valor_Nodo; { Información asociada a cada nodo }
        Existe    : Boolean;
        Visitado  : Boolean;
        Lista_Arcos: Punt_Arco;
    End;

    Grafo = Record
        Nodos: Array [ Indice ] Of Nodo;
    End;
```

Escribir un subprograma en Pascal que partiendo del nodo que contiene un cierto carácter **x**, recorra en anchura el grafo (recorrido BFS). El prototipo del subprograma será:

```
Procedure BFS ( gra: Grafo, x: Char );
```

Escribe los prototipos de los subprogramas auxiliares vistos en clase que consideres necesario utilizar.