



1.- ¿Qué muestra por pantalla el siguiente segmento de código que trabaja con listas representadas con punteros

```
Type
  Puntero = ^Nodo;

  Nodo = Record
    Info: Integer;
    Sig : Puntero;
  End;

Procedure XXX ( punt: Puntero )
Begin
  If ( punt <> NIL ) Then
  Begin
    If ( punt^.Info <= 5 ) Then
      XXX ( punt^.Sig );

      Write ( punt^.Info );

    If ( punt^.Info > 5 ) Then
      XXX ( punt^.Sig )
  End
End;
```

```
Var
  list, paux: Puntero;
  i: Integer

Begin
  list := NIL;
  For i := 1 To 9 Do
  Begin
    New ( paux );
    paux^.Info := i;
    paux^.Sig := list;
    list := paux
  End;

  XXX ( list )
End.
```

Muestra:

- 2.- Supongamos un grafo representado mediante matrices de adyacencia y mediante listas de adyacencia.
- La representación mediante matrices de adyacencia aporta mayor cantidad de información que la representación con listas de adyacencia.
 - La representación mediante listas de adyacencia aporta mayor cantidad de información que la representación con listas de adyacencia.
 - La información aportada en las dos representaciones es la misma, pero es más fácil obtenerla en la representación mediante matrices de adyacencia que mediante listas de adyacencia.
 - la única representación que aporta toda la información sobre el grafo es la representación mediante matrices dispersas.
- 3.- Supongamos un árbol binario de búsqueda con 255 nodos.
- ¿Cuál es la altura máxima del árbol?
 - ¿Y la altura mínima?
 - ¿Cuál es el número máximo de hojas (nodos cuyos dos enlaces a hijos están a NIL)?
 - ¿Y el número mínimo de hojas?
 - ¿Cuál es número de enlaces a NIL si tenemos 8 hojas?
 - ¿Y si tenemos 128 hojas?
- 4.- Supongamos un montículo de máximos con 255 nodos.
- ¿Cuál es la altura máxima del árbol?
 - ¿Y la altura mínima?
 - ¿Cuál es el número máximo de hojas (nodos cuyos dos enlaces a hijos están a NIL)?
 - ¿Y el número mínimo de hojas?
 - ¿Cuál es número de enlaces a NIL si tenemos 8 hojas?
 - ¿Y si tenemos 128 hojas?

Apellidos:	Nombre:
Cognoms:	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:

- **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

5.- ¿Por qué es interesante la utilización de tipos abstractos de datos en programación?

- a.- Para poder utilizar unidades en Turbo Pascal.
- b.- Para separar claramente la implementación y representación de datos de su utilización.
- c.- Para que los programas que escribamos sean más fáciles de compilar y ejecutar.
- d.- Para reducir el número de instrucciones de los algoritmos y mejorar su eficiencia.

6.- Supongamos el T.A.D. lista...

- a.- La inserción y eliminación de elementos en la representación con punteros es más costosa que en la representación mediante *arrays*.
- b.- En la representación con *arrays* y cursores (índices adicionales que nos indiquen la secuencia de elementos dentro del vector), la inserción y eliminación es similar a la inserción y eliminación en la representación con punteros.
- c.- Las listas no pueden representarse con *arrays* de ninguna forma.
- d.- La representación con *arrays* siempre es una ventaja frente a la representación con punteros.

7.- En general...

- a.- ...la mayor o menor complejidad de las representaciones depende en gran manera del número de elementos que contenga la estructura.
- b.- ...la complejidad espacial (ocupación de memoria) de las estructuras representadas con *arrays* es menor que la representación con punteros.
- c.- ... la complejidad espacial (ocupación de memoria) de las estructuras representadas con punteros es menor que la representación con *arrays*.
- d.- ...no se puede hacer ningún tipo de afirmación porque muchas de los T.A.D. no es posible representarlos con *arrays*.

8.- ¿Cuál será la ocupación de memoria de un grafo de 5 nodos, de un máximo de 10, y 3 relaciones...

a.- ...si lo representamos mediante matrices de adyacencia.	b.- ...si lo representamos mediante listas de adyacencia.
-------------------------------------------------------------	-----------------------------------------------------------

Supondremos que la información contenida en los nodos y en las relaciones son números reales (4 bytes). Los punteros ocupan 4 bytes y los enteros 2 bytes.

9.- Supongamos la existencia de la función sobre pilas:

Concatenar (Pila, Pila) -> Pila

Que pone las dos pilas pasadas como parámetros una detrás de la otra y devuelve la pila resultante.

¿Qué hace la función 'f', aplicada sobre *sta* de tipo Pila?

f (Ø) → Ø (f aplicado sobre la pila vacía devuelve la pila vacía)

f (sta) → Concatenar (f (Desapilar (sta)), Apilar (Ø, CimaPila (sta)))



P.1.- Escribir un procedimiento ITERATIVO que libere la memoria ocupada por un árbol binario representado mediante punteros. La declaración de tipos y el prototipo de la función son los siguientes.

```
Type
  Arbol = ^Nodo;

  Nodo = Record
    Info: Valor;
    Izdo, Dcho: Arbol;
  End;

Procedure BorrarIterativo ( Var arbo: Arbol );
```

Escribe el prototipo de todas aquellas operaciones y tipos de datos que creas oportuno utilizar.

P.2.- Supongamos una lista representada con punteros doblemente enlazada. Realizar un procedimiento en Pascal que elimine de la lista aquellos elementos mayores que un cierto 'x' pasado como parámetro y los inserte en una cola. Escribe los prototipos de aquellas funciones sobre colas que utilices.

El prototipo de la función será la siguiente:

```
Type
  Puntero = ^Nodo;

  Nodo = Record
    Info: Integer;
    Ante, Sigü: Puntero;
  End;

Procedure Eliminar ( Var list: ListaD, x: Integer, Var que: Cola );
```

P.1.- Escriu un procediment ITERATIU en Pascal que allibere la memòria ocupada per un arbre binari representat mitjançant punters. La declaració de tipus i el prototipus de la funció son les següents.

```
Type
  Arbol = ^Nodo;

  Nodo = Record
    Info: Valor;
    Izdo, Dcho: Arbol;
  End;

Procedure BorrarIterativo ( Var arbo: Arbol );
```

Escriu els prototips de totes aquelles operacions i tipus de dades que cregues oportu utilitzar.

P.2.- Supposem una llista representada amb punters doblement enllaçada. Realitzar un procediment en Pascal que elimine de la llista aquells elements majors que un cert 'x' pasat com paràmetre i els inserte en una cua. Escriu els prototipus de aquelles funcions sobre cues que utilitzes.

El prototipus de la funció serà el següent:

```
Type
  Puntero = ^Nodo;

  Nodo = Record
    Info: Integer;
    Ante, Sigü: Puntero;
  End;

Procedure Eliminar ( Var list: ListaD, x: Integer, Var que: Cola );
```