

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED enero 2006

Modalidad 0

- Normas:**
- La entrega del test no corre convocatoria.
 - Tiempo para efectuar el test: **35 minutos**.
 - Una pregunta mal contestada elimina una correcta.
 - Las soluciones al examen se dejarán en el campus virtual.
 - **Una vez empezado el examen no se puede salir del aula hasta finalizarlo.**
 - En la **hoja de contestaciones** el verdadero se corresponderá con la **A**, y el falso con la **B**.

	V	F		
En la especificación de un TAD, las operaciones auxiliares son visibles para los usuarios.	<input type="checkbox"/>	<input type="checkbox"/>	1.	F
Una operación del TAD X que tenga la sintaxis Crear() →X es una operación constructora generadora.	<input type="checkbox"/>	<input type="checkbox"/>	2.	V
En C++, los miembros <i>protected</i> son privados para el exterior, pero permiten el acceso a las clases derivadas.	<input type="checkbox"/>	<input type="checkbox"/>	3.	V
Dadas las clases <i>TDir</i> y <i>TVectorDir</i> : <pre>class TDir { public: private: int e1; char c1; }; class TVectorDir { public: private: TDir *vector; int dim; }; TVectorDir::~TVectorDir () { if (v!=NULL) delete v; dim=0; v=NULL; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	4.	F
¿Es correcta la implementación del destructor de <i>TVectorDir</i> ?	<input type="checkbox"/>	<input type="checkbox"/>	5.	F
El nivel de un nodo en un árbol coincide con la longitud del camino desde la raíz a dicho nodo	<input type="checkbox"/>	<input type="checkbox"/>	6.	V
Dado un único recorrido de un árbol binario lleno es posible reconstruir dicho árbol	<input type="checkbox"/>	<input type="checkbox"/>	7.	F
Sea el tipo <i>arbin</i> definido en clase. La semántica de la operación <i>nodos</i> es la siguiente: <pre>Var i,d:arbin; x:item; nodos(crear_arbin())=0 nodos(enraizar(i,x,d))=nodos(i)+nodos(d)</pre>	<input type="checkbox"/>	<input type="checkbox"/>	8.	F
A los árboles generales también se les llama árboles multicamino de búsqueda	<input type="checkbox"/>	<input type="checkbox"/>	9.	V
El ítem medio (según la relación de orden) almacenado en un árbol binario de búsqueda lleno siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	10.	V
Un árbol completo siempre está balanceado respecto a la altura	<input type="checkbox"/>	<input type="checkbox"/>	11.	F
En un árbol AVL siempre que se inserte una etiqueta hay que realizar una rotación	<input type="checkbox"/>	<input type="checkbox"/>	12.	V
El coste temporal en el peor caso de la operación de inserción en un árbol 2-3-4 es $\log_2(n+1) \approx \log_2(n)$ siendo “n” el número total de ítems	<input type="checkbox"/>	<input type="checkbox"/>	13.	V
Se puede obtener un único árbol 2-3-4 a partir de su recorrido por niveles	<input type="checkbox"/>	<input type="checkbox"/>	14.	V
La semántica de la operación <i>anterior</i> vista en clase es la siguiente: <pre>VAR L1: lista; x: item; p: posicion; anterior(L1, primera(L1)) = error_posicion(); si p != ultima(L1) entonces anterior(L1, siguiente(L1, p)) = p anterior(inscabeza(L1, x), primera(L1)) = primera(inscabeza(L1, x))</pre>	<input type="checkbox"/>	<input type="checkbox"/>	15.	F
Sea el tipo <i>vector</i> definido en clase. La semántica de la operación <i>recu</i> es la siguiente: <pre>Var v:vector; i,j:int; x:item; recu(crear_vector(),i)=error_item() si i<>j entonces recu(asig(v,i,x),j)=recu(v,j) sino recu(asig(v,i,x),j)=TRUE</pre>	<input type="checkbox"/>	<input type="checkbox"/>	16.	V
<i>crear_pila()</i> , <i>apilar(pila,item)</i> y <i>desapilar(pila)</i> son operaciones constructoras del tipo <i>pila</i> .	<input type="checkbox"/>	<input type="checkbox"/>	17.	F
Todo árbol binario de búsqueda es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	18.	F
La semántica de la operación <i>Base</i> que actúa sobre una <i>pila</i> y devuelve el primer elemento apilado es la siguiente: <pre>Base(crear_pila ()) = crear_pila () Base(apilar(crear_pila (), x)) = x Base(apilar(p, x)) = Base(p)</pre>	<input type="checkbox"/>	<input type="checkbox"/>		