

Apellidos:

Nombre:

Convocatoria:

DNI:

Examen PED febrero 2007

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 C++ y cuando se emplea composición (<i>layering</i>), los métodos de la clase derivada pueden acceder a la parte pública de la clase base.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
En C++, el puntero <i>this</i> no se puede emplear para modificar el objeto al que apunta.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
En C++, los constructores se pueden invocar explícitamente cuando el programador lo desee (por ejemplo: <i>TLista a; a.TLista();</i>).	<input type="checkbox"/>	<input type="checkbox"/>	3	F
En C++, la siguiente declaración es incorrecta: $int\& a = 1;$	<input type="checkbox"/>	<input type="checkbox"/>	4	V
En la escala de complejidades se cumple que $O(\log n) \subset O(\log \log n)$.	<input type="checkbox"/>	<input type="checkbox"/>	5	F
El algoritmo de búsqueda binaria estudiado en clase (búsqueda de un elemento en un vector ordenado) tiene una complejidad de $\mathcal{O}(1)$.	<input type="checkbox"/>	<input type="checkbox"/>	6	V
Para un vector de naturales, se define la operación <i>eliminar</i> que borra las posiciones pares del vector marcándolas con "0" (para calcular el resto de una división, se puede utilizar la operación MOD). La sintaxis y la semántica de la operación <i>eliminar</i> es la siguiente: <i>eliminar</i> : $vector \rightarrow vector$ <i>Var</i> v:vector; i: entero; x:natural; <i>eliminar</i> (<i>crear_vector</i> ()) = <i>crear_vector</i> () <i>si</i> (i MOD 2) == 0 <i>entonces</i> <i>eliminar</i> (<i>asignar</i> (v,i,x)) = <i>asignar</i> (<i>eliminar</i> (v),i,0) <i>si no</i> <i>eliminar</i> (<i>asignar</i> (v,i,x)) = <i>asignar</i> (<i>eliminar</i> (v),i,x)	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La semántica de la operación <i>ultima</i> vista en clase es la siguiente: <i>VAR</i> L1: lista; x: item; <i>si</i> <i>esvacia</i> (L1) <i>entonces</i> <i>ultima</i> (<i>inscabeza</i> (L1, x)) = <i>primera</i> (L1) <i>si no</i> <i>ultima</i> (<i>inscabeza</i> (L1, x)) = <i>ultima</i> (L1)	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Es posible reconstruir un único árbol binario de altura 6 a partir de un recorrido en preorden con 62 etiquetas.	<input type="checkbox"/>	<input type="checkbox"/>	9	F
La sintaxis y semántica de la operación <i>quita_hojas</i> , que actúa sobre un árbol binario y devuelve el árbol binario original sin sus hojas, es la siguiente: <i>quita_hojas</i> (<i>arbin</i>) \rightarrow <i>arbin</i> <i>VAR</i> i, d: <i>arbin</i> ; x: <i>item</i> ; <i>quita_hojas</i> (<i>crea_arbin</i> ()) = <i>crea_arbin</i> () <i>quita_hojas</i> (<i>enraizar</i> (<i>crea_arbin</i> (), x, <i>crea_arbin</i> ())) = <i>crea_arbin</i> () <i>quita_hojas</i> (<i>enraizar</i> (i, x, d)) = <i>enraizar</i> (<i>quita_hojas</i> (i), x, <i>quita_hojas</i> (d))	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Dados los recorridos de preorden, postorden y niveles de un árbol binario de altura 2 y 1 hoja es posible reconstruir 2 árboles binarios.	<input type="checkbox"/>	<input type="checkbox"/>	11	V
Todo árbol AVL es un árbol 2-3-4	<input type="checkbox"/>	<input type="checkbox"/>	12	F
La operación (<i>DIVIDEHIJODE2</i> (p, q)) en la inserción de un elemento en un árbol 2-3-4 puede aumentar la altura del árbol original.	<input type="checkbox"/>	<input type="checkbox"/>	13	F
En el algoritmo del borrado de un elemento en un árbol 2-3-4 si q es 2-nodo y r es 3-nodo hay que hacer una ROTACIÓN.	<input type="checkbox"/>	<input type="checkbox"/>	14	V