

Apellidos, Nombre:

DNI:

Examen PED marzo 2015

Modalidad 0

Normas:

- Tiempo para efectuar el test: **20 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 algebraica, las operaciones constructoras se clasifican en generadoras y modificadoras.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
Las ecuaciones (vistas en clase) que permiten realizar la suma de números naturales son las siguientes: VAR x, y: natural; suma(x, cero) = x suma(cero, x) = x suma(x, suc(y)) = suma(x, y)	<input type="checkbox"/>	<input type="checkbox"/>	2	F
Dentro de la especificación algebraica de los números naturales, definimos la sintaxis de la función F como: F: natural → BOOL, y su semántica como: F(cero)=TRUE, F(suc(cero))=FALSE, F(suc(suc(x)))=F(x). Para el número natural x=35, la función F devolvería FALSE. Nota: se asume que x=35 es la forma simplificada de indicar x=suc(suc(suc(.....suc(cero).....))).	<input type="checkbox"/>	<input type="checkbox"/>	3	V
Todo árbol binario de altura 9 y 511 nodos es un árbol binario lleno y además es árbol binario de búsqueda.	<input type="checkbox"/>	<input type="checkbox"/>	4	F
Sea el método Primera perteneciente a la clase TLista que devuelve la primera posición de la lista que lo invoca: <pre>TPosicion TLista::Primera() { TPosicion p; p.pos = primero; return p; }</pre> <pre>class TLista { public: ... private: TNode *primero; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	5	F
En el método Primera, se invoca de forma implícita a los constructores de TPosicion y TLista.				
En C++, si la variable p es un puntero a un objeto, entonces la expresión p.f() es sintácticamente correcta.	<input type="checkbox"/>	<input type="checkbox"/>	6	F
La complejidad temporal del siguiente fragmento de código es O(n) int i, j, n, sum; for (i = 4; i < n; i++) { for (j = i-3, sum = a[i-4]; j <= i; j++) sum += a[j]; cout << "La suma del subarray " << i-4 << " es " << sum << endl; }	<input type="checkbox"/>	<input type="checkbox"/>	7	V
La mejor complejidad temporal que se puede conseguir en un algoritmo es O(n), siendo "n" la talla del problema.	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Es posible reconstruir un único árbol binario de búsqueda a partir de su recorrido en postorden	<input type="checkbox"/>	<input type="checkbox"/>	9	V
El máximo número de nodos en un nivel i-1 de un árbol binario es 2^{i-2} , $i \geq 2$	<input type="checkbox"/>	<input type="checkbox"/>	10	V
Un camino en un árbol es una secuencia a_1, \dots, a_s de árboles tal que para todo $i \in \{1, \dots, s-1\}$, a_{i+1} es subárbol de a_i .	<input type="checkbox"/>	<input type="checkbox"/>	11	V
El grado de un árbol es el máximo nivel que pueden tener sus subárboles.	<input type="checkbox"/>	<input type="checkbox"/>	12	F
La operación desencolar vista en clase es la siguiente: VAR c: cola, x: item; desencolar(crear()) = crear() si esvacía(c) entonces desencolar(encolar(c, x)) = crear() si no desencolar(encolar(c, x)) = encolar(desencolar(c), x)	<input type="checkbox"/>	<input type="checkbox"/>	13	V
El ítem medio (según la relación de orden en la búsqueda) almacenado en un árbol binario de búsqueda siempre se encuentra en la raíz.	<input type="checkbox"/>	<input type="checkbox"/>	14	F