

Examen PED marzo 2018

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 de los números naturales vista en clase, si no definimos la semántica de la operación <i>suma</i> , las ecuaciones: $\text{suc}(\text{suc}(\text{cero}))$ y $\text{suma}(\text{cero}, \text{suc}(\text{cero}))$ podrían denotar diferentes valores.	<input type="checkbox"/>	<input type="checkbox"/>	1	V
En C++, la herencia y el layering son el mismo concepto.	<input type="checkbox"/>	<input type="checkbox"/>	2	F
En C++, si una clase no contiene el destructor, un objeto creado nunca se destruye hasta que finaliza el programa.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
La expresión de la complejidad temporal es equivalente a la talla o tamaño del problema.	<input type="checkbox"/>	<input type="checkbox"/>	4	F
La función POT_2 descrita a continuación presenta una complejidad temporal $O(2^n)$ función POT_2 (n: natural): natural opción n = 1: devuelve 2 n > 1: devuelve 2 * POT_2(n-1) fopción ffunción	<input type="checkbox"/>	<input type="checkbox"/>	5	F
La función de búsqueda BINARIA de un elemento en un vector ordenado tiene una complejidad temporal en su peor caso de $O(\log_{10} n)$	<input type="checkbox"/>	<input type="checkbox"/>	6	F
Dada la operación <i>Examen</i> definida como sigue, donde IC equivale a inscabeza: <i>Examen</i> (Lista) → Lista VAR L1:Lista; x,y:ítem <i>Examen</i> (crear_lista())=crear_lista() <i>Examen</i> (IC(crear_lista(), x)) = IC(crear_lista(),x) <i>Examen</i> (IC(IC(L1,x)y))=IC(IC(<i>Examen</i> (L1),y),x) Si se aplica sobre la Lista=(a,b,c,d), en la que a es el primer elemento de la lista, se obtiene como resultado la Lista=(b,a,d,c)	<input type="checkbox"/>	<input type="checkbox"/>	7	V
Un árbol binario con un único nodo es un árbol lleno.	<input type="checkbox"/>	<input type="checkbox"/>	8	V
En la especificación algebraica del tipo Lista visto en clase, la sintaxis y la semántica de la operación de borrado de un elemento (<i>borrar</i>) según la posición es la siguiente: <i>borrar</i> (lista, posicion) → lista VAR L1: lista; x: ítem; p: posicion; <i>borrar</i> (crear(), p) = crear() si p == primera(inscabeza(L1, x)) entonces <i>borrar</i> (inscabeza(L1, x), p) = L1 si no <i>borrar</i> (inscabeza(L1, x), p) = inscabeza(<i>borrar</i> (L1, p), x)	<input type="checkbox"/>	<input type="checkbox"/>	9	V
Teniendo en cuenta la sintaxis y la semántica de las operaciones del tipo Cola vistas en clase, se cumpliría la siguiente ecuación: VAR a,b,c: ítem; desencolar(desencolar(encolar(encolar(encolar(crear(),a),b),c))) = c	<input type="checkbox"/>	<input type="checkbox"/>	10	F
En la representación SECUENCIAL del tipo Cola visto en clase utilizando tipos base ("arrays"), tanto para colas (array normal) como para colas circulares (array circular), la condición de cola vacía es <code>tope == fondo</code> .	<input type="checkbox"/>	<input type="checkbox"/>	11	F
En la especificación algebraica de los números naturales vista en clase, la sintaxis y la semántica de la operación de multiplicación (<i>mult</i>) es la siguiente: <i>mult</i> : natural natural → natural VAR x, y: natural; <i>mult</i> (cero, x) = cero <i>mult</i> (x, cero) = cero <i>mult</i> (y, x) = suma(<i>mult</i> (suc(y), x), x)	<input type="checkbox"/>	<input type="checkbox"/>	12	F
Grado de un árbol es el número máximo de hijos que pueden tener sus subárboles (si el árbol es n-ario, el grado es n).	<input type="checkbox"/>	<input type="checkbox"/>	13	V