

Apellidos:

Nombre:

Convocatoria:

DNI:

## Examen PED febrero 2009

### Modalidad 0

- Normas:**
- La entrega del test **no** corre convocatoria.
  - Tiempo para efectuar el test: **25 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++, al declarar una clase "A" como AMIGA de otra clase "B", todas las funciones miembro de "B" automáticamente pasan a ser funciones AMIGAS de "A"	<input type="checkbox"/>	<input type="checkbox"/>	1	F
En C++, si una clase "B" se construye por composición (layering), a partir de otra clase "A", definiendo un objeto miembro de la clase "A" en su parte privada, al invocar al constructor de "B" se invoca antes al constructor de "A" y luego al de "B"	<input type="checkbox"/>	<input type="checkbox"/>	2	V
Las funciones y clases amigas se tienen que declarar en la parte pública de la clase.	<input type="checkbox"/>	<input type="checkbox"/>	3	F
Para el siguiente algoritmo, la complejidad temporal en su peor caso sería O(1):  <pre>for (i=0; i&lt;100; i++)   for (j=0; j&lt;100; j++)     if (v[i]&lt;v[j]) v[i]=v[j];     else v[j]=v[i];</pre>	<input type="checkbox"/>	<input type="checkbox"/>	4	V
La complejidad temporal en su peor caso del siguiente fragmento de código es O(n)  <pre>int i, j, n, sum; for (i = 4; i &lt; n; i++) {   for (j = i-3, sum = a[i-4]; j &lt;= i; j++) sum += a[j];   cout &lt;&lt; "La suma del subarray " &lt;&lt; i-4 &lt;&lt; " es " &lt;&lt; sum &lt;&lt; endl; }</pre>	<input type="checkbox"/>	<input type="checkbox"/>	5	V
La semántica de la operación esvaciapos del tipo vector vista en clase es la siguiente:  <pre>VAR v: vector; i, j: int; x: item; esvaciapos( crear( ), i ) = CIERTO esvaciapos( asig( v, i, x ), j )   si ( i == j ) entonces FALSO   si no esvaciapos( v, j ) fsi</pre>	<input type="checkbox"/>	<input type="checkbox"/>	6	V
La semántica de la operación anterior 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 ) ) = L1</pre>	<input type="checkbox"/>	<input type="checkbox"/>	7	F
La sintaxis y semántica de la operación simétricos, que comprueba que 2 árboles binarios son simétricos, es la siguiente:  <pre>simétricos(arbin, arbin) → bool VAR i1, d1, i2, d2: arbin; x, y: item; simétricos(enraizar(i1, x, d1), crea_arbin( )) = FALSO simétricos(crea_arbin( ), enraizar(i1, x, d1)) = FALSO simétricos(enraizar(i1, x, d1), enraizar(i2, y, d2)) =   si ( x == y ) entonces ( simétricos(i1, d2) &amp; simétricos( d1, i2) )   sino FALSO</pre>	<input type="checkbox"/>	<input type="checkbox"/>	8	F
Un árbol binario completo es un AVL	<input type="checkbox"/>	<input type="checkbox"/>	9	F