

Alumno:

**CUESTIONARIO:** Para cada consigna indicar la opción correcta, que en todos los casos siempre será 1 (UNA). (20 pts)

<p><b>1)</b></p> <p>En relación a la <b>recursividad</b>, indicar cuál de las afirmaciones es incorrecta:</p> <p>.....</p>	<p>a) Una función es recursiva si se llama a sí misma, con un conjunto de datos más pequeño que el original cada vez.</p> <p>b) La recursión divide el problema en dos partes: la que resuelve el problema (caso base) devolviendo un resultado y la que no sabe cómo resolverlo (caso general) y realiza llamadas recursivas.</p> <p>c) Las funciones recursivas pueden emplear parámetros por referencia.</p> <p>d) Si se omite el caso base o la llamada recursiva se realiza de manera incorrecta, impidiendo llegar al caso base, se produce una recursividad infinita.</p>
<p><b>2)</b></p> <p>En relación a las <b>estructuras dinámicas</b>, indicar cuál de las afirmaciones es correcta:</p> <p>.....</p>	<p>a) Son estructuras que crecen y se reducen durante la ejecución del programa.</p> <p>b) La sentencia delete ptr; elimina el puntero ptr que referencia a un área en la memoria principal.</p> <p>c) Para ordenar una lista enlazada dinámica, los nodos deben moverse a su posición correcta.</p> <p>d) El puntero al último nodo de una lista enlazada se conoce como puntero maestro.</p>
<p><b>3)</b></p> <p>En relación a los <b>tipos de datos y operadores</b>, y considerando las variables enteras <math>a=8</math>, <math>b=17</math>, <math>c=-5</math>, indicar cuál de las siguientes afirmaciones da como resultado false:</p> <p>.....</p>	<p>a) <math>\text{sqrt}(-b)*(b/a)==c*(-1)+3</math>;</p> <p>b) <math>!(3*c++ != -(b==2)-(a/4))</math>;</p> <p>c) <math>\text{sqrt}(\text{pow}(c,2))&lt;=\text{pow}(a++/4,3)</math>;</p> <p>d) <math>!(\text{abs}(c+a) &gt; b \text{ or } \text{false and } (++a*c) &lt; \text{pow}(c,3))</math>;</p>
<p><b>4)</b></p> <p>En relación a los <b>programas en C++</b>, indicar cuál de las siguientes afirmaciones es correcta:</p> <p>.....</p>	<p>a) El programa ejecutable deberá obtenerse cada vez que éste necesite ejecutarse/correrse.</p> <p>b) Los errores lógicos (errores del programador) pueden ser detectados por el programa compilador.</p> <p>c) Se denomina programa fuente al archivo de texto escrito utilizando un programa depurador.</p> <p>d) El programa fuente, escrito en C++, se traduce a lenguaje de máquina mediante un programa compilador.</p>
<p><b>5)</b></p> <p>Al hablar de <b>complejidad</b>, indicar cuál es la afirmación correcta:</p> <p>.....</p>	<p>a) Obtener el mayor valor de las últimas 3 columnas de una matriz de NxM valores enteros tiene complejidad <math>O(N^2)</math>.</p> <p>b) Intercambiar el último y penúltimo nodos de una lista enlazada ordenada tiene complejidad <math>O(1)</math>.</p> <p>c) Obtener el n-ésimo valor de Fibonacci de manera recursiva tiene complejidad <math>O(N^2)</math>.</p> <p>d) Obtener la intersección entre dos listas enlazadas y ordenadas de N elementos</p>

	enteros cada una, tiene complejidad $O(N^2)$ .
<p>6)</p> <p>En relación a las <b>cadenas</b> en C++, indicar cuál de las afirmaciones es incorrecta:</p> <p>.....</p>	<p>a) <code>char cad1[30],cad2[30]="Oid mortales"; strncpy(cad1,cad2,3);</code></p> <p>b) <code>const string cad3="Las provincias unidas"; string cad4=cad3+" del sur"; cad3+='.';</code></p> <p>c) <code>cout&lt;&lt;cad3.substr(4)&lt;&lt;endl;</code></p> <p>d) <code>char cad5[20]; cin&gt;&gt;cad5; for(unsigned i=0; i&lt;strlen(cad5);i++){ cout&lt;&lt;char(toupper(cad5[i])); }</code></p>
<p>7)</p> <p>En relación a las <b>funciones</b>, indicar cuál de las afirmaciones es incorrecta:</p> <p>.....</p>	<p>a) En la llamada a una función debe haber un argumento (parámetro actual) para cada parámetro (formal) en la definición de la función.</p> <p>b) El pasaje de parámetros por copia a una función implica utilizar parámetros de salida como únicos argumentos.</p> <p>c) Con funciones que no devuelven un resultado (tienen un tipo de valor de retorno void), el control regresa a la función llamadora cuando la función llamada alcanza su llave de fin de función o ejecuta la instrucción return;</p> <p>d) Un parámetro por referencia es un alias para su correspondiente argumento en la llamada a una función, y se indica colocando un signo &amp; después del tipo de dato del parámetro en el prototipo/cabecera de la función.</p>
<p>8)</p> <p>En relación a las <b>cadenas</b> en C++, indicar cuál de las afirmaciones es incorrecta:</p> <p>.....</p>	<p>e) <code>char cad1[30],cad2[30]="Oid mortales"; strncpy(cad1,cad2,3);</code></p> <p>f) <code>const string cad3="Las provincias unidas"; string cad4=cad3+" del sur"; cad3+='.';</code></p> <p>g) <code>cout&lt;&lt;cad3.substr(4)&lt;&lt;endl;</code></p> <p>h) <code>char cad5[20]; cin&gt;&gt;cad5; for(unsigned i=0; i&lt;strlen(cad5);i++){ cout&lt;&lt;char(toupper(cad5[i])); }</code></p>
<p>9) En relación a la <b>estructura repetitiva if</b>, indicar cuál será la salida del siguiente código:</p> <pre>int a=17, b=3; if(a%b&lt;2) {     if (b&lt;a) cout &lt;&lt; pow(b,4) &lt;&lt; endl; else cout &lt;&lt; sqrt(--a) &lt;&lt; endl;     if ((a-=14)&lt;(b*=2))         cout &lt;&lt; ++a &lt;&lt; endl; }</pre> <p>.....</p>	<p>a) 4</p> <p>b) 81</p> <p>c) 4 3</p> <p>d) 81 3</p>
<p>10)</p> <p>En relación a los <b>arreglos</b> en C++, indicar cuál de las afirmaciones es correcta:</p> <p>.....</p>	<p>a) Son estructuras estáticas que se declaran en tiempo de compilación.</p> <p>b) Pueden contener otros arreglos como elementos.</p> <p>c) Su tamaño físico puede indicarse en tiempo de ejecución.</p> <p>d) Pueden devolverse a través de parámetros de salida indicados de la siguiente manera: nombre-funcion(tipo-dato &amp; nombre-parametro);</p>

**Responder V o F y justificar su respuesta:**

- a) **(2 pts)** Sea z una referencia a una variable entera (por ejemplo: `int a= 10; int & z = a;`); z tiene una dirección de memoria asignada, al igual que lo tiene a.
- b) **(2 pts)** Para leer un valor entero de un archivo binario "datos.bin", se debe hacer:
- ```
int a;  
ifstream f;  
f.open("datos.bin");  
f >> a;
```

**Responder brevemente:**

- a) **(3 pts)** Indicar qué son las pilas y las colas. Describir características de cada una.
- b) **(3 pts)** Explicar el uso de structs. Ejemplificar con código.

**Problema OmegaUp:**

<https://omegaup.com/arena/problem/Desplazamientos-y-algo-mas/>