

## Algoritmos y Estructuras de Datos. K1025. 1er Recuperatorio 2da Evaluación Parcial.

Fecha: 20/11/2019

Apellido y nombre: ..... Legajo: .....

Para aprobar debe sumar al menos 60 puntos (calificación 6), siendo 80 puntos el mínimo para aprobación directa (calificación 8).

**1) Algoritmo.** Seleccione la respuesta correcta e ingrese un comentario explicando su decisión.

<pre>void miFuncion(Nodo*&amp; u, Elem e) {     Nodo* n = new Nodo();     n-&gt;elem = e;     n-&gt;sig = u;     n-&gt;ant = u-&gt;ant;     Nodo* a = u-&gt;ant;     u-&gt;ant = n;     a-&gt;sig = n; }</pre>	<p>¿Qué resultado produce esta función?</p> <p><input type="checkbox"/> Agrega el elemento <b>e</b> al inicio de la estructura</p> <p><input type="checkbox"/> Agrega el elemento <b>e</b> al final de la estructura</p> <p><input type="checkbox"/> Ordena los nodos según el elemento <b>e</b></p> <p><input type="checkbox"/> Ninguna de las anteriores</p>
<p>Comentario/supuestos sobre la respuesta seleccionada:</p> <p>.....</p> <p>.....</p>	

(20 puntos)

**2) Atrás.** Un nuevo navegador web mobile necesita incorporar la opción de ir “atrás” (sitio anterior) como parte de sus funcionalidades básicas. Internamente el navegador agrega a una variable de tipo string la dirección (URL) que se visita y el objetivo es guardar en memoria el historial de todas las direcciones visitadas, de forma tal que esté disponible si el usuario selecciona la opción de ir atrás.

Se pide: Implementar esta funcionalidad para que al invocar desde **main()** a la función **atras()**, ésta garantice que retorne la dirección de la última página visitada y las sucesivas anteriores. También crear la definición de la función **navegar()** que recibe la URL y guardará en memoria temporal la página a visitar. Explicar desde **main()** con: “www.youtube.com”, “www.smn.gob.ar” y “www.slack.com”. Luego invocar 2 veces **atras()** mostrando cuál sería el retorno en cada llamada (agregar comentarios en el código fuente). Crear las estructuras necesarias.

(30 puntos)

**3) Servidores.** Una plataforma de videojuegos on-line mantiene en memoria todos los servidores de todos los videojuegos disponibles en tiempo real. La plataforma puede gestionar hasta 256 títulos (juegos) diferentes y por cada uno mantiene un listado de servidores disponibles. Cada registro de servidor posee los siguientes datos:

idJuego	nombreJuego	idServidor	nombreServidor	ping	cantJugadores
(entero 0-255)	(array de car. [15])	(entero)	(array de car. [15])	(entero)	(entero)

donde **ping** es la latencia actual del servidor en milisegundos.

Se pide: Cargar la estructura de datos con los registros de servidores a leer desde un archivo invocando a la función **cargarServidores()**, que recibirá por parámetro la estructura, la ruta al archivo y deberá agregar los servidores leídos ordenados ascendentemente por el campo **ping**. Si la estructura ya tiene ese servidor, deberá reemplazarlo. También crear la función **borrarLentos()**, que recibirá por parámetro la estructura, una cifra en milisegundos y borrará todos los servidores que superen el valor de ping recibido por parámetro. Crear las estructuras necesarias.

(50 puntos)