

**Algoritmos y Estructuras de Datos. K1041. 1<sup>er</sup> Recuperatorio de 2<sup>do</sup> Examen Parcial.**

**Fecha: 23/11/2018**

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

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

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

<pre>template &lt;typename T&gt; T miFuncion(Nodo&lt;T&gt;* &amp; p) {     Nodo&lt;T&gt;* aux = p;     T v = aux-&gt;info;     p = aux-&gt;sig;     delete aux;     return v; }</pre>	<p>¿Qué resultado produce esta función?</p> <p><input type="checkbox"/> Elimina un nodo de una estructura pila</p> <p><input type="checkbox"/> Elimina un nodo de una estructura lista</p> <p><input type="checkbox"/> Elimina un nodo de una estructura cola</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) Checkin.** Se desea conocer el tiempo de espera promedio de los pasajeros para hacer el checkin en el aeropuerto. Cada vez que ingresa un nuevo pasajero a la fila (única), se emite y coloca en el equipaje un código de barras autoadhesivo con el id de pasajero. Una vez que el pasajero es atendido, se lee este código y el sistema almacena el tiempo que transcurrió hasta que salió de la fila. Puede haber más de un mostrador de atención operativo.

Se pide: Crear un programa que gestione el ingreso de nuevos pasajeros desde la función **nuevoPasajero()**, donde se genera el registro Pasajero:

id_pasajero	momento_de_ingreso
Entero	entero largo

Que al ser atendido, la lectura del código invoca la función **Pasajero atenderProximo()**, con cuyo retorno se deberá calcular la diferencia de tiempo invocando la función **long tiempoDeEspera(long momento)** que retornará el tiempo transcurrido respecto del dato recibido por parámetro. En la función **main()** se acumula el tiempo de espera de todos los pasajeros y la cantidad. Crear las estructuras necesarias. **Info:** Para obtener el momento actual, invocar la función **time(NULL)** de la biblioteca **time.h**

(40 puntos)

**3) Logística.** Cada unidad de una flota de transporte de carga está equipada con un dispositivo que registra cada 60 minutos las mediciones de todos los sensores instalados y las almacena en una estructura que contiene registros de tipo Medición, cuyos atributos son:

id_unidad	fecha_hora	velocidad	temperatura	combustible	ubicaciones
entero	entero largo	decimal	decimal	decimal	array de LatLon [5]

A su vez, cada registro almacena en la estructura **ubicaciones**, la posición geográfica medida cada 10 minutos en registros LatLon:

latitud	longitud
decimal	decimal

Se pide: Crear la función **descargarMediciones()** que recibe por parámetro la estructura con mediciones, la ruta al archivo y que deberá guardar en el archivo todas las mediciones ordenadas por fecha\_hora. Crear las estructuras necesarias. **Info:** El valor de fecha\_hora se obtuvo invocando a la función **time(NULL)**.

(40 puntos)