

Reglas del juego:

- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.

1. (2.0) Elabore el código del método intercambiar(int posición, int vecino):bool de una lista doblemente encadenada, de acuerdo a las siguientes especificaciones:

```
public bool intercambiar(int posición, int vecino)  
    Miembro de estructuras.ListaDoble
```

Resumen:

Permite intercambiar el elemento de la colección indicado por el parámetro posición, con su vecino izquierdo o derecho, de acuerdo a lo que determine el parámetro vecino.

Parámetros:

posición: Posición del elemento a intercambiar en la lista doblemente encadenada

vecino: Vecino con el cuál se hará el intercambio: 0 si es el vecino izquierdo y 1 si es el vecino derecho

Valores devueltos:

true si se pudo efectuar el intercambio y *false* en caso contrario (cuando el vecino con el que se va a intercambiar no sea válido)

Nota: Recuerde que no está permitido el intercambio de la información contenida dentro de los nodos de la lista.

2. (1.5) Robin Hood y sus acompañantes han sido capturados por los soldados del Reino, y saben que serán condenados a la horca por los robos realizados a importantes miembros de la monarquía. Sabiendo esto, el grupo de capturados ha decidido que prefieren suicidarse antes que morir en manos del Rey. De acuerdo a esto, han decidido formar un círculo e ir matándose cada tres ladrones, hasta que todos estén muertos. Elabore el código en C# que permita poner n ladrones en el círculo e ir matando cada m ladrones, retornando al final el último ladrón que logre sobrevivir. Indique para lo anterior qué estructuras y clases (0.5) utilizará en la solución del problema.
Código: 1.0
Diagrama de clases: 0.5
3. (1.5) Le han solicitado elaborar una aplicación que permita convertir un número decimal a base octal, utilizando el encadenamiento del módulo de las divisiones sucesivas. Elabore en C# un método **no recursivo** que calcule el nuevo número, teniendo en cuenta que únicamente puede utilizar estructuras tipo listas, colas o pilas para la solución. De acuerdo a la estructura escogida, justifique claramente la razón para preferir dicha estructura frente a las otras.
Código: 1.0
Justificación: 0.5