

## Práctica integradora (1er parcial)

- Si bien el presente no es un examen, puede ser beneficioso intentar realizarlo dentro de un marco de tiempo equivalente al que tendrá en el mismo, en un ambiente sin interrupciones y de forma individual en papel.
- También recomendamos aspirar a escribir de forma prolija, o pasar el contenido de borradores a hojas finales para poder estimar correctamente durante el proceso los tiempos que esta tarea demandaría durante el examen.
- Los procedimientos y funciones primitivos que son dados en el examen deberán ser implementados en caso de que se desee probar la solución en máquina, por lo que recomendamos que aspire a realizar esto solamente una vez terminado de realizar los ejercicios en papel.
- Aconsejamos leer el enunciado completo antes de empezar a resolver los ejercicios, ya que ayuda a comprender mejor el dominio.
- Recuerde que el parcial será a libro abierto, que puede consultar cualquier material (en papel) que haya sido escrito antes de comenzar el examen y usar sin definir todas las funciones y procedimientos de la biblioteca. Tenga esto en cuenta al realizar su solución.
- Pensar bien la estrategia a seguir, expresando la misma mediante procedimientos y funciones. Recordar escribir primero los contratos.

### La Convención de Motoqueros

La Convención de Motoqueros es una simulación del comportamiento de los motoqueros a lo largo de una convención. En las convenciones, a los motoqueros les gusta interactuar con otros motoqueros y asociarse a clubes.

La Convención de Motoqueros se puede modelar en Gobstones. Cada motoquero puede representarse con una celda del tablero. Un motoquero es vecino de otro si sus correspondientes celdas son lindantes en cualquiera de las 4 direcciones ortogonales. La imagen de la izquierda de la Figura 1 muestra una convención posible.

Durante una convención, se arman clubes. Es decir, grupos de motoqueros que están conectados entre sí. Dos motoqueros se dicen conectados cuando hay una secuencia de motoqueros vecinos que los unen.

Podemos además asignarles números a cada club, para diferenciarlos unos de otros y determinar cosas como qué grupo tiene más miembros, o cuál tiene

menos. La imagen de la Figura 1 muestra una convención de motoqueros, mientras que a su derecha se muestra la misma convención con los diversos grupos numerados, habiendo asignado un número a cada grupo.

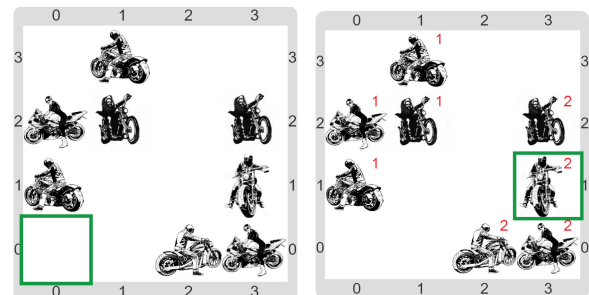


Figura 1: A la izquierda, una convención de motoqueros, con 8 motoqueros en total. A la derecha, la misma convención con los clubes marcados. Hay 2 clubes distintos en la convención, siendo ambos clubes de 4 miembros.

Se cuenta con los procedimientos y funciones primitivas que se enuncian a continuación:

- **elMotoqueroPerteneceAUnClub()**  
PROPÓSITO: Indica si el motoquero actual pertenece a algún club.  
PRECONDICIÓN: Hay un motoquero en la celda actual.  
TIPO: Booleano.
- **clubAlQuePerteneceUnMotoquero()**  
PROPÓSITO: Describe el número del club al cual pertenece el motoquero actual.  
PRECONDICIÓN: Hay un motoquero en la celda actual y pertenece a un club.  
TIPO: Número.
- **IncluirEnElClubNúmero\_(númeroDeClub)**  
PROPÓSITO: Incluye al motoquero actual como perteneciente al club con el número dado.  
PRECONDICIÓN: Hay un motoquero en la celda actual y el mismo no pertenece todavía a ningún club.  
PARÁMETROS: **númeroDeClub** - Número. El número de club en el cual incluir al motoquero
- **hayMotoqueroAcá()**  
PROPÓSITO: Indica si hay un motoquero en la celda actual.  
PRECONDICIÓN: Ninguna.  
TIPO: Booleano.

## EJERCICIOS

**Ejercicio 1)** Realizar la función `cantidadDeClubesEnLaConvención()` que determine la cantidad total de clubes de motoqueros que hay en la convención.

Para ello se propone una estrategia *top-down* para contar los clubes que consiste en recorrer motoqueros sin club, asignando un número de club distinto y consecutivo a cada uno de los motoqueros sin club y todos los motoqueros conectados a este. El último número de club asignado debería permitir determinar la cantidad de clubes totales. Para esto se propone realizar las siguientes funciones y procedimientos (aunque pueden requerirse o ser útil algunos adicionales):

**Ejercicio a)** La función `hayMotoqueroSinClub()` que indica si en la convención hay algún motoquero que todavía no tenga un club.

**Ejercicio b)** El procedimiento `IrAMotoqueroSinClub()` que, asumiendo que hay un motoquero sin club, deja el cabezal sobre el mismo.

**Ejercicio c)** El procedimiento `AsignarClub_AMotoquerosConectados(númeroDeClub)` que dado un número de club, asigna al motoquero actual y a todos los motoqueros conectados a este, como pertenecientes al dicho club. Se puede asumir que estos motoqueros no tienen un club asignado.

Este problema es a su vez complejo, y se propone la siguiente estrategia para solucionarlo. Notar que algunas subtarefas en los ítems a continuación pueden ser útiles para solucionar este problema.

- Paso 1, marcar al motoquero actual como perteneciente al club dado.
- Paso 2, buscar en la convención algún motoquero sin club que tenga como vecino un motoquero del club dado e ir hasta el mismo.
- Repetir sucesivamente los pasos 1 y 2 hasta que no haya más motoqueros en la convención que sean vecinos de otros con el club dado.

**Ejercicio d)** La función `hayMotoqueroSinClubConVecinoEn_(númeroDeClub)` que indica si en la convención hay algún motoquero sin club todavía que tenga de vecino otro motoquero que tenga por club `númeroDeClub`.

**Ejercicio e)** `IrHastaElMotoqueroSinClubConVecinoEn_(númeroDeClub)` que posiciona el cabezal sobre un motoquero que no tenga club todavía y que tenga por vecino un motoquero en el club `númeroDeClub`.

**Ejercicio 2)** Escribir la función `clubConMásMotoqueros()` que determina cuál es el club que tiene más motoqueros en la convención. Se propone como estrategia marcar primero los clubes, y luego contabilizar la cantidad de miembros de cada club, para luego saber cuál tiene más. De esto surgen los siguientes procedimientos y funciones (nuevamente, realizar adicionales puede resultar útil):

**Ejercicio a)** El procedimiento `AsignarClubATodosLosMotoqueros()` que asigna un club a cada motoquero activo de la convención, y que sigue una estrategia similar a la del ejercicio 1.

**Ejercicio b)** La función `cantidadDeMiembrosDelClub_(númeroDeClub)` que dado un número de club, describe la cantidad de miembros que posee el club dado.