

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA Y ESTRUCTURA DE DATOS**

**2da. práctica (tipo B)**  
**(Primer Semestre 2025)**

Duración: 1h 50 min.

- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías `iostream`, `iomanip`, `limits`, `cstring`, `cmath` o `fstream`
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_LAB2_P#&` (donde # representa el número de la pregunta a resolver y & la alternativa)

---

**Pregunta 1 (10 puntos)**

Durante la gran carrera de cuadrigas (carro tirado por cuatro caballos de frente) en la película *Ben-Hur*, el juez principal del evento observa un problema en el sistema de cronometraje: el orden de llegada de las cuadrigas en un punto intermedio de la pista fue mal registrado. La información fue almacenada como una **lista simplemente enlazada**, en la que cada nodo representa una cuadriga con su respectivo número de inscripción (ID), el nombre del conductor, y el color de su equipo.

Para evitar confusiones antes del veredicto oficial, el juez exige reordenar la lista según el siguiente criterio **sin utilizar estructuras auxiliares como nuevas listas, arreglos o vectores**:

- Todas las cuadrigas con **ID par** deben aparecer al inicio de la lista.
- Todas las cuadrigas con **ID impar** deben ubicarse al final.
- Dentro de cada grupo (pares e impares), **debe mantenerse el orden original relativo** de llegada.

Por ejemplo:

**Lista Inicial:**

[ID: 17, Nombre: Messala, Equipo: Rojo]

[ID: 4, Nombre: Ben-Hur, Equipo: Azul]

[ID: 12, Nombre: Artax, Equipo: Verde]

[ID: 7, Nombre: Drusus, Equipo: Negro]

### Lista Final:

[ID: 4, Nombre: Ben-Hur, Equipo: Azul]  
[ID: 12, Nombre: Artax, Equipo: Verde]  
[ID: 17, Nombre: Messala, Equipo: Rojo]  
[ID: 7, Nombre: Drusus, Equipo: Negro]

Se le pide:

- Definir las estructuras necesarias que soporten el enunciado descrito, así como el ingreso de los datos de una lista de cuadrigas. Recuerde que esta lista debe estar desordenada. **(2.0 puntos)**
- Implementar una función que reordene la lista siguiendo la lógica indicada. No puede utilizar memoria extra (nodos nuevos) de ningún tipo ni arreglos, vectores, estructuras auxiliares o archivos. Puede crear los punteros auxiliares que necesite. **Solo puede emplear iterativas simples (7.0 puntos)**
- Implementar una función que imprima las listas en el formato indicado **(1.0 punto)**

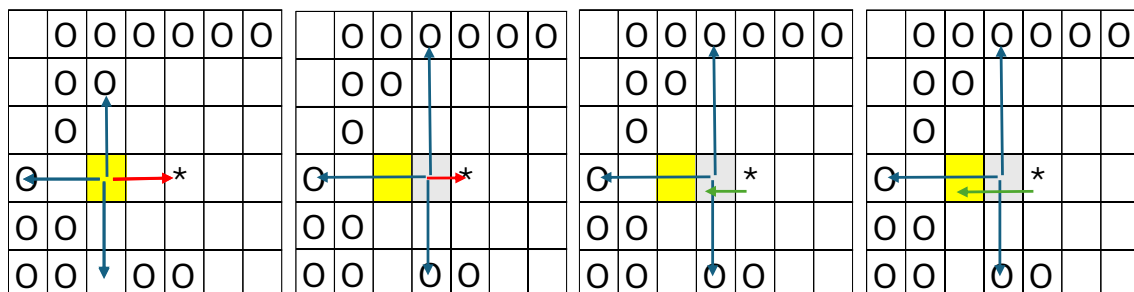
### Pregunta 2 (10 puntos)

Una empresa dedicada a la minería ha adquirido un robot para la detección de vetas de oro. Esta unidad robótica se envía a una determinada coordenada (X,Y) dentro de una mina para iniciar sus labores de detección en un espacio NxM. Por ahora este robot solo puede ir en una sola dirección para la detección de vetas de oro, puede ir hacia arriba, abajo, derecha o izquierda. Además, durante su tarea de detección busca oro en horizontal y vertical pero no en diagonal. Se sabe que el robot tiene una cantidad de combustible limitada así que solo puede operar mientras tenga una cantidad que le ayude a moverse entre **C** posiciones. Otro detalle es que el robot se puede encontrar con una roca muy dura y no puede continuar su búsqueda, por tal motivo si aún tiene combustible cambia a la dirección contraria con respecto a la seguida originalmente. Se sabe que el robot detecta el inicio de una veta y alerta a los mineros enviándole las coordenadas del inicio. A continuación, un ejemplo de la búsqueda que realiza el robot:

**N = 6 M = 7 C = 4**

El robot se envía a la posición: **(3,2)**

Movimiento de la sonda: **derecha**.



Como se observa en el mapa hay una roca en la ubicación (3,4). El llegar y salir de la roca también consume combustible.

**El resultado de la búsqueda será:**

Oro: 1 2  
Oro: 3 0  
Oro: 5 3  
Oro: 0 3  
Oro: 5 3  
Oro: 0 3  
Oro: 1 2

Si le ponemos  $C=6$

**El resultado de la búsqueda será:**

Oro 1 2  
Oro 3 0  
Oro 5 3  
Oro 0 3  
Oro 5 3  
Oro 0 3  
Oro 1 2  
Oro 4 1  
Oro 2 1  
Oro 4 0

Implemente un programa que, utilizando una sola función recursiva, devuelva las posiciones de inicio de las vetas de oro. Solo puede emplear una matriz de entrada. No puede usar iterativas dobles, arreglos o matrices auxiliares. Puede representar los datos dentro de la matriz como números si lo desea (10 puntos).

Al finalizar el laboratorio, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este laboratorio.

Profesores del curso:

Ana Roncal  
Fernando Huamán  
David Allasi  
Rony Cueva  
Heider Sanchez

San Miguel, 03 de mayo del 2025