

Padrón:

Apellido y Nombre:

Punteros)

```

/*
 * Indicar la salida y la memoria que libera el GC.
 * Datos:
 * '@' es 64
 * 'A' es 65
 */
public static void main(String[] args) {
    int A, C, F = 0;
    char G;
    int H;
    int[] L = new int[4];
    int[] M = L;
    H = 75;
    G = 'B';
    A = H;
    M[0] = 30;

    while (L[0] > 10) {
        H--;
        A--;
        C = A;
        System.out.println(C + "-" + H + "-" + M[1] + "-" + F);
        F = A;
        C--;
        M[0] -= 30;
    }
    L[0] = A++;
    M[1] = L[0];
    System.out.println(A + "-" + G + "-" + M[0] + "-" + L[1]);

    G = (char) H;
    M[2] = L[0] + A;
    System.out.println(G + "-" + H + "-" + M[1] + "-" + L[2]);
}

```

TDA) Implementar un TDA que modele el **Tablero de resultado** de Ajedrez. El tablero se crea a partir de una cantidad de jugadores. Debe seguir las siguientes reglas:

- Cada jugador solo puede jugar contra cada uno de los otros jugadores 2 partidas, una con blancas y otra con negras, hacer un método para recibir la puntuación.
- Cada partido puede tener el resultado GANAN BLANCAS, TABLAS (empate) o GANAN NEGRAS.
- Un juez puede anular una partida y no suma puntos.
- Cada jugador tiene un número del 1 al total.
- Conocer si termino el torneo.
- Dado un numero de jugador, devolver el puntaje, siendo 3 puntos por partido ganado, 1 por empate y resta 1 punto por partida anulada.
- Devolver el jugador ganador.

LISTA) Dado el listado de Socios de un club, filtrar un socio por familia cuyo grupo familiar no tubo atrasos en las cuotas para asignar un descuento.

Implementar el método buscarSociosParaDescuento de la clase AdministradorDeClubSocial a partir de la siguiente especificación de su interfaz:

```

class AdministradorDeClubSocial {
    public:
        /**
         * post: devuelve una nueva Lista con los Socios, uno por familia, cuyo grupo familiar no tuvo deudas.
         */
        public Lista<Socio> buscarSociosParaDescuento (Lista<Socio> socios);
}

class Socio {
    public boolean tuvoDeuda();
    public String getGrupoFamiliar();
}

```

Los alumnos que tienen aprobado algún punto, no deben hacerlo.

Para aprobar es necesario tener al menos el 60% de los ejercicios correctos y completos.

Todas las implementaciones deben respetar la teoría de TDA.

Duración del examen: 2 horas y 30 minutos