

Padrón:

Apellido y Nombre:

Mail:

Puntero) Indicar la salida

<pre> /* 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[6]; int[] M = L; H = 69; G = 'B'; A = H; 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]); </pre>	<pre> while (L[0] > 10) { H--; A--; C = A; System.out.println(C + "-" + H + "-" + M[1] + "-" + F); F = A; C--; M[0] -= 30; } //Que hace el garbage collector </pre>
---	--

TDA) Implementar una clase que modele el juego del Ahorcado. Un Ahorcado debe ser creada con la palabra a adivinar y la cantidad de errores permitidos. El Ahorcado debe poseer métodos para:

- Conocer la cantidad de letras que tiene la palabra.
- Conocer el estado del juego: JUGANDO, GANADO, PERDIDO.
- Proponer una letra, devolviendo la cantidad de apariciones que tiene esa letra en la palabra a adivinar. Si la letra no está en la palabra se incrementa la cantidad de errores cometidos. Sólo se pueden arriesgar letras mientras que está en estado JUGANDO. Cuando la cantidad de errores cometidos es mayor que la cantidad de errores permitidos, el juego queda PERDIDO.
- Conocer la cantidad de errores cometidos.
- Arriesgar una palabra, devolviendo si es correcta o no. Si es correcta el juego queda GANADO y en caso contrario PERDIDO.
- Conocer la palabra cuando el juego está GANADO o PERDIDO, si está JUGANDO debe devolver una cadena vacía.

LISTA) Un Blog está compuesto por un conjunto de Artículos. Cada Artículo posee un título, un texto y una lista de palabras excluyentes.
Implementar el método buscarArticulos de la clase Blog a partir de la siguiente especificación de su interfaz:

```

class Blog {

    public:
        /**
         * post: el Blog queda creado sin Artículos.
         */
        Blog();
        /**
         * post: devuelve los Artículos que componen el Blog.
         */
        Lista<Articulo> getArticulos();
        /**
         * post: devuelve una nueva Lista con los Artículos que tengan en su titulo o texto todas las palabras
         claves (en cualquier orden), pero ninguna debe estar prohibida
         */
        Lista<Articulo> buscarArticulos(Lista<String> palabrasClaves);
}

```

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