

INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por el número de lista seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.java, Ex08frp2.java, etc.**
- ➔ En los comentarios de cada programa se debe indicar el nombre completo, la fecha y el turno.

EJERCICIOS

1. Realiza una nueva versión del juego “Busca el tesoro” realizado en clase con los siguientes cambios:
 - Hay tres minas en lugar de una. Si el jugador pisa una mina, está herido, y si pisa otra más, está muerto.
 - Sigue habiendo un único tesoro. Si el jugador lo encuentra, gana.
 - No pueden coincidir las coordenadas de ninguno de los elementos, ya sean minas o tesoro.
 - Para que el ejercicio se pueda corregir con facilidad, se deben mostrar por pantalla, a modo de chuleta, las coordenadas de todos los elementos (minas y tesoro).

2. Realiza una función que devuelva el número complementario a otro que se pasa como parámetro. Cada dígito del número complementario es la diferencia con 9 del dígito original. Por ejemplo el complementario de 308566 es el 691433 y el complementario de 92491 es el 7508. Prueba la función desde un programa. A continuación se muestra la cabecera de la función:

```
public static long complementario(long x)
```

3. Se dice que una matriz cuadrada – tiene el mismo número de filas que de columnas - es triangular superior cuando todos los valores que están por debajo de la diagonal principal son cero. Por ejemplo, la siguiente matriz es triangular superior:

```
2 7 5
0 4 1
0 0 8
```

Realiza una función que diga si una matriz dada es o no triangular superior y pruébala en un programa. Recuerda que si la matriz no es cuadrada, se puede concluir directamente que no es triangular superior, sin hacer ningún otro tipo de comprobación. La cabecera de la función es la siguiente:

```
public static boolean esTriangularSuperior(int[ ][ ] a)
```

4. Una cadena de multicines nos ha encargado una aplicación para la venta on-line de entradas. Como parte de esa aplicación, tenemos que implementar una función que coloque a los espectadores en una fila de butacas. Una fila de butacas es un array de una dimensión. Cada celda se corresponde a una butaca que tiene el valor 0 si está libre y el valor 1 si está ocupada. La función recibe como parámetros el array con la información de la fila de butacas y un número que es la cantidad de gente que se quiere colocar en esa fila. Si se puede colocar a los espectadores con éxito, la función devolverá un 0 y si no se puede, un -1. Lo primero que tiene que hacer la función es comprobar si hay sitio, si no lo hay devolverá -1. Si hay sitio, primero intentará colocar juntos a todo el grupo y, si no puede, los irá colocando en los primeros huecos que encuentre.

Por ejemplo, si $a = \{ 0, 1, 0, 0, 0, 1, 0 \}$ y se quieren colocar 3 nuevos espectadores, la función devolverá 0 y el array se habrá modificado quedando así $a = \{ 0, 1, 1, 1, 1, 1, 0 \}$.

Prueba la función desde un programa. La cabecera de la función es la siguiente:

```
public static int ocupa(int[ ] fila, int usuarios)
```