



La situación se representa en una matriz llamada **iconosAula**, donde a cada persona sentada en el aula le corresponde un icono. A ésta la que acompaña otra matriz llamada **nombresAula**, que contiene los nombres de las personas situadas en cada posición. también tienes una función que permite imprimir ambas matrices por consola.

Adminístrate bien el tiempo: Si un trozo del programa no te sale, avanza al siguiente ejercicio sin esa parte si es posible, hay varios ejercicios que puedes hacer sin hacer los anteriores. **Te recomiendo que conforme hagas una función, la pruebes en el main para asegurarte de que funciona bien. Luego borra la prueba y deja el main como se te pide en el examen, pero no vayas a hacer medio examen sin probar nada: Mala idea.**

Para hacer este programa, sigue los ejercicios:

1 - **(0,5 Puntos)** Crea una función llamada `intercambiarPosiciones`, que reciba por argumentos una matriz bidimensional de String, y cuatro bytes: `fila1`, `columna1`, `fila2`, `columna2`. En su interior, intercambiará los valores de la posición (`fila1,columna1`) de la matriz, con los de la posición (`fila2,columna2`).

2 - **(0,5 Puntos)** Crea una función llamada `posicionAleatoriaMatriz`. Devolverá un String, y recibirá dos argumentos: un byte llamado `tamañoFilas`, y otro byte llamado `tamañoColumnas`. En su interior, usará la clase `Random` para sortear dos números: Uno llamado `fila`, que podrá valer entre 0 y `tamañoFilas-1`, y otro llamado `columna`, que podrá valer entre 0 y `tamañoColumnas-1`. Por último, la función devolverá los dos números concatenados en un String, y separados por una coma.

*Ejemplo: Si a la función se le pasa 6 como `tamañoFilas` y 12 como `tamañoColumnas`, devolverá un string con dos números separados por coma. El primero será un aleatorio que puede valer entre 0 y 5, y el segundo un aleatorio que puede valer entre 0 y 11. Por ejemplo, podría devolver el String "3,7"*

3 - **(0,25 Puntos)** El malvado caballo no se fía de sus alumnos, por lo que va a hacer que algunos de ellos se cambien el sitio. Para ello, crea una función llamada `intercambiarAlumnos`, que no devuelva nada, y reciba por argumentos dos matrices bidimensionales de String: Una llamada `iconos` y otra llamada `nombres`. También recibirá por argumentos un byte llamado `veces`. En su interior, hará un bucle que itere tantas veces como indique el argumento `veces`. Para cada una de esas veces:

- A. **(0,5 Puntos)** Se llamará a la función `posicionAleatoriaMatriz`, pasándole por argumentos el número de filas y columnas de la matriz `iconos`. Al String resultante hazle `split` en la coma. El array resultante contendrá en la posición 0 la fila sorteada (la llamaremos `f1`), y en la posición 1 la columna sorteada (la llamaremos `c1`).
- B. **(0,25 Puntos)** Se comprobará que en la posición (`f1,c1`) de la matriz `iconos` recibida por argumentos, hay uno de estos iconos: 🧐, 🧐 o 🐎. De no ser así, se volverá a repetir el paso A tantas veces como sea necesario, hasta que la posición sorteada contenga uno de esos tres valores.
- C. **(0,25 Puntos)** Se hará lo mismo que en los puntos A y B otra vez, sorteando otra fila y otra columna (las llamaremos `f2` y `c2`) a partir de la función `posicionAleatoriaMatriz`, en la que nos aseguraremos que hay un 🧐, 🧐 o 🐎.
- D. **(0,25 Puntos)** Usa la función `intercambiarPosiciones` para intercambiar los valores de (`f1,c1`) y (`f2,c2`) tanto en la matriz `iconos` como en la matriz `nombres`.

4 - **(0,25 Puntos)** Desde el main, tras lo que ya hay puesto, pide por teclado el número de intercambios que quieres que el caballo haga. Usa ese número y las matrices para llamar a la función `intercambiarAlumnos`. Después, vuelve a imprimir las dos matrices usando la función `imprimeAula`.

5 - **(0,5 Puntos)** Haz una función llamada `buscarCaballo`, que devuelva un byte, y reciba por argumentos una matriz de String. Es un pre-requisito para esta función, que el caballo debe estar en la columna 2 de alguna de las filas en la matriz, o esta función no funcionará bien. No tienes que comprobarlo dentro del código, pero ten en cuenta esto para documentar la función, que es el último ejercicio. En el interior de la función, recorre la matriz, en la dirección que quieras, buscando qué fila contiene un caballo en la columna 2. Una vez la encuentre, la devolverá.

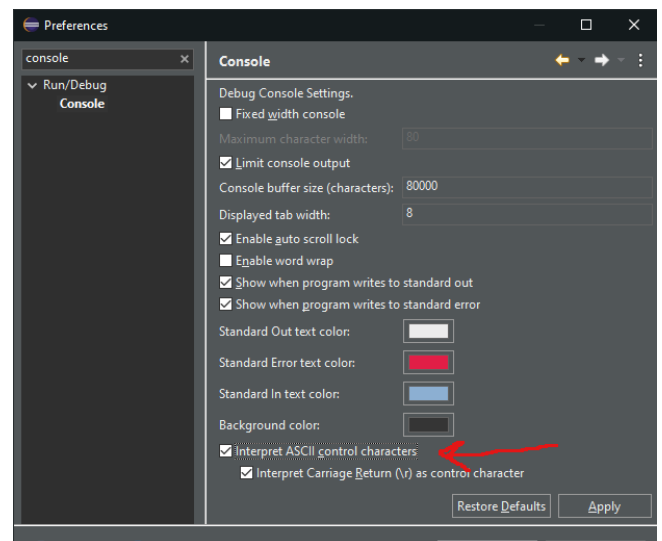
6 - **(0,15 Puntos)** Haz una función llamada moverCaballo, que devuelva un boolean, y reciba por argumentos dos matrices de String (llamadas iconos y nombres) y un byte llamado dirección. Es un pre-requisito de la función que el byte dirección valga 0 1 o -1. No es necesario que compruebes esto dentro de la función, pero tenlo en cuenta a la hora de documentarla, que es el último ejercicio. En el interior de la función:

- **(0,25 Puntos)** Primero llamará a la función buscarCaballo, usando como argumento cualquiera de las dos matrices de String que tienes disponible. Guarda el resultado de llamar a buscarCaballo en una variable llamada posCaballo.
- **(0,75 Puntos)** Después, si dirección es 1, usa la función intercambiarPosiciones para intercambiar la posición (posCaballo,2) por la posición (posCaballo+1,2) tanto en la matriz iconos como en la matriz nombres. Si dirección es -1, usa la función intercambiarPosiciones para intercambiar la posición (posCaballo,2) por la posición (posCaballo-1,2) tanto en la matriz iconos como en la matriz nombres.
- **(0,15 Puntos)** Comprueba (usando el método que quieras) si en la posición (0,2) o en la posición (iconos.length-1,2) hay un caballo. En caso afirmativo devuelve true. En caso negativo, devuelve false.

7 - **(0,15 Puntos)** En el main, tras lo hecho en el ejercicio 4, pide por teclado un número (byte) llamado movimientos. Después inicializa un byte llamado direccion a 1. Tras esto, haz un bucle que cuente de 0 a movimientos, y en su interior haga lo siguiente:

- **(0,15 Puntos)** Llamar a moverCaballo, usando las dos matrices que tienes en el main, y la variable dirección como argumentos.
- **(0,25 Puntos)** Solo si moverCaballo devuelve true, cambiar el signo a la variable dirección: Si vale 1 que cambie a -1, si vale -1 que cambie a 1.
- **(0,15 Puntos)** Volver a imprimir ambas matrices usando la función imprimeAula que te he dado.
- **(No puntúa - Opcional)** Pega este código al final del bucle. No tienes que entender lo que hace: Sólo pégalo. Hará que pase un segundo entre impresión e impresión de la matriz, y se limpie la pantalla entre ellas (pero solo si tienes Eclipse 2022-12 y jdk 19 o superiores y la opción de interpretar caracteres de control ASCII habilitada en opciones. La puedes habilitar en Window -> Preferences ):

```
try {
    Thread.sleep(1500);
} catch (InterruptedException e) {
    e.printStackTrace();
}
for(byte i=0;i<iconosAula.length;i++) {
    System.out.println("\r");
}
```



8 - **(1,25 Puntos)** Haz que el programa pueda recibir los argumentos -i seguido de un número, y -m seguido de un número. Los argumentos deben poder estar o no estar presentes, y ser usados en cualquier orden. Si se recibe -i, tomar como número de intercambios el valor recibido a continuación, y no pedirlo por teclado como pide el ejercicio 4. Si se recibe -m, tomar como número de movimientos el valor recibido a continuación, y no pedirlo por teclado como pide el ejercicio 7.

9 - **(0,5 Puntos)** Documenta correctamente todo el código del proyecto.