

No se pueden usar clases Java como las de Fecha y Hora, Arrays, clase String, Colecciones, etc.

Realiza un Proyecto Java desde Eclipse de nombre **RA3_ApellidosNombre**. Crea una clase para cada ejercicio, el nombre es: **EjercicioX**, donde X es el número del ejercicio. Para leer los datos numéricos en los ejercicios CONTROLAR LA EXCEPCIÓN **InputMismatchException**. Si lo leído no es un número se debe leer de nuevo.

Enero 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Febrero 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	Marzo 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Abril 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	Mayo 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Junio 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Julio 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Agosto 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Septiembre 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Octubre 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	Noviembre 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	Diciembre 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

1º) Partimos del calendario del año 2021. Sabemos que el día 1 de Enero del 2021 fue Viernes. Se trata de averiguar el nombre del día de la semana para un día y un mes.

Realiza una clase Java de nombre **Ejercicio1.java** en el que crees una función de nombre **NombreDiaSemana** que reciba un día y un mes, de tipo **int**; y devuelve un mensaje en el que aparezca el día, nombre del mes y el nombre del día de la semana que es. Desde **main()** harás las llamadas a la función (**no se leerá por teclado**). Por ejemplo, para estas llamadas:

```
int dia = 1; int mes = 2;
System.out.println(NombreDiaSemana(dia, mes));
dia = 4; mes = 5;
System.out.println(NombreDiaSemana(dia, mes));
```

La salida debe ser:

El día 1 de Febrero es Lunes.

El día 4 de Mayo es Martes.

2º) El Mastermind es un juego que consiste en adivinar una combinación numérica de cuatro cifras en la que es importante la posición de cada cifra. Realiza un programa Java de nombre **Ejercicio2.java** que simule este juego. En el programa se debe definir una combinación numérica de 4 cifras, sin repetir ninguna, y estas cifras serán las que se tienen que adivinar en la menor cantidad de intentos posibles.

El jugador tiene **10 intentos** para adivinar la combinación, cada intento consiste en introducir por teclado 4 números (de 0 a 9) **sin repetir ninguno**, y una vez introducidos comprobarlos con la combinación a adivinar. Si hay alguno repetido se muestra un mensaje y se vuelven a leer de nuevo. Hay que averiguar:

1. Los **ACIERTOS**, o **MUERTOS**, es la cantidad de dígitos que introdujo el jugador que coinciden en la combinación y están en la misma posición;
2. Las **COINCIDENCIAS**, o **HERIDOS**, es la cantidad de dígitos que introdujo el jugador que también están en la combinación pero en una posición distinta.

Por ejemplo, esta es la combinación de números que hay que adivinar **[0, 9, 3, 6]**:

- El jugador introduce: **[1, 2, 3, 4]**, el programa debe mostrar: **Un acierto** ya que el 3 está en la misma posición que en la combinación original y **cero coincidencias**, ya que el 1, 2 y 4 no coinciden con ningún número de la combinación original.
- El jugador introduce: **[2, 2, 4, 3]**, el programa debe mostrar: que **hay números repetidos** y por tanto no se muestran ni aciertos ni coincidencias.
- El jugador introduce: **[5, 6, 3, 0]**, el programa debe mostrar: **Un acierto** ya que el 3 está en la misma posición que en la combinación original y **dos coincidencias**, ya que el 6 y el 0 coinciden con el 0 y el 6 de la combinación original pero no están en la misma posición.
- El jugador introduce: **[0, 6, 3, 9]**, el programa debe mostrar: **Dos aciertos** ya que el 0 y el 3 están en la misma posición que en la combinación original y **dos coincidencias**, ya que el 6 y el 9 coinciden con el 6 y el 9 de la combinación original pero no están en la misma posición.
- El jugador introduce: **[0, 9, 3, 7]**, el programa debe mostrar: **Tres aciertos** ya que el 0, el 9 y el 3 están en la misma posición que en la combinación original y **cero coincidencias**, ya que el 7 no está en la combinación original.
- El jugador envía: **[0, 9, 3, 6]**, el programa debe mostrar: **Cuatro aciertos** ya que el 0, el 9, el 3 y el 6 están en la misma posición que en la combinación original y; **cero coincidencias**.

El juego terminará cuando se adivine la combinación o cuando el número de intentos sea mayor de 10. Ejemplo de ejecución, suponiendo que la combinación a adivinar es: [0, 9, 3, 6], la salida es la siguiente:

```

Intento : 1
  Introduce número (0 y 9) 1: 1
  Introduce número (0 y 9) 2: 2
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 4
  Números introducidos: [1, 2, 3, 4]
  Aciertos: 1
  Coincidencias: 0
Intento : 2
  Introduce número (0 y 9) 1: 2
  Introduce número (0 y 9) 2: 2
  Introduce número (0 y 9) 3: 4
  Introduce número (0 y 9) 4: 3
  Hay Números repetidos: [2, 2, 4, 3] leerlos de nuevo
Intento : 3
  Introduce número (0 y 9) 1: 5
  Introduce número (0 y 9) 2: 6
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 0
  Números introducidos: [5, 6, 3, 0]
  Aciertos: 1
  Coincidencias: 2
Intento : 4
  Introduce número (0 y 9) 1: 0
  Introduce número (0 y 9) 2: 6
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 9
  Números introducidos: [0, 6, 3, 9]
  Aciertos: 2
  Coincidencias: 2
Intento : 5
  Introduce número (0 y 9) 1: 0
  Introduce número (0 y 9) 2: 9
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 7
  Números introducidos: [0, 9, 3, 7]
  Aciertos: 3
  Coincidencias: 0
Intento : 6
  Introduce número (0 y 9) 1: 0
  Introduce número (0 y 9) 2: 9
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 3
  Hay Números repetidos: [0, 9, 3, 3] leerlos de nuevo
Intento : 7
  Introduce número (0 y 9) 1: 0
  Introduce número (0 y 9) 2: 9
  Introduce número (0 y 9) 3: 3
  Introduce número (0 y 9) 4: 6
  Números introducidos: [0, 9, 3, 6]
  Aciertos: 4
  Coincidencias: 0
<<ENHORABUENA, HAS ACERTADO LA COMBINACIÓN EN 7 INTENTOS>>

```

NO USAR
ARRAYS NI
CADENAS

CONTROLAR LA
EXCEPCIÓN
`InputMismatchException`

3º) Realiza un programa Java de nombre **Ejercicio3.java** que pinte montañas. Se pedirá por teclado en un proceso repetitivo el número de filas que tiene la montaña y el número de montañas que se tienen que mostrar; el proceso repetitivo finalizará cuando el número de filas sea 0. Para pintar las montañas el número de montañas debe ser > que 0, si no se cumple se vuelve a leer. Ejemplo de ejecución:

Numero de filas de la montaña: 3 Numero de montañas: 2 <pre> * * * * * * * ** *</pre>	Numero de filas de la montaña: 4 Numero de montañas: 5 <pre> * * * * * * * * * * * * * * * * * * * * * * * * * * ** ** ** ** ** *</pre>
---	--

Numero de filas de la montaña: 4
 Numero de montañas: 0
 Numero de montañas: -3
 Numero de montañas: 4

```

*      *      *      *
* *    * *    * *    * *
*   *   *   *   *   *   *
*     **     **     **     *
```

Numero de filas de la montaña: 1
 Numero de montañas: 1
 *
 Numero de filas de la montaña: 0

Numero de filas de la montaña: 6
 Numero de montañas: 6

```

*      *      *      *      *      *
* *    * *    * *    * *    * *    * *
*   *   *   *   *   *   *   *   *   *
*     **     **     **     **     **     **
*       *      *      *      *      *      *
```

4º) Realiza una clase Java de nombre **Ejercicio4.java**. Se debe introducir por teclado dos números, el primero es un número de tipo **long** y el segundo es el número a buscar dentro del anterior y debe ser un número entre 0 y 9, de tipo **short**, si no está entre 0 y 9 se debe leer de nuevo. El programa tiene que mostrar en qué posiciones aparece el segundo número leído por teclado en el primer número leído por teclado empezando a contar las posiciones de izquierda a derecha. Ejemplo de ejecución:

Nombre y Apellidos.

Introduzca un número: 123456789

Introduzca el dígito a buscar (0 a 9): 4

Si no está entre 0 y 9 se lee de nuevo.

Posiciones en las que se encuentra el dígito 4, en el numero 123456789
 Posiciones: 4

=====

Introduzca un número: 1145336673

Introduzca el dígito a buscar (0 a 9): 3

Posiciones en las que se encuentra el dígito 3, en el numero 1145336673
 Posiciones: 5, 6, 10

=====

Introduzca un número: 11

Introduzca el dígito a buscar (0 a 9): 1

Posiciones en las que se encuentra el dígito 1, en el numero 11
 Posiciones: 1, 2

=====

Introduzca un número: 1000

Introduzca el dígito a buscar (0 a 9): 0

Posiciones en las que se encuentra el dígito 0, en el numero 1000
 Posiciones: 2, 3, 4

=====

Introduzca un número: 0

El proceso finalizará cuando el primer número leído sea 0. Si el primer número es menor que 0 se pasa a positivo. Si el dato introducido no es numérico, se pinta mensaje indicándolo y se lee de nuevo. Ejemplo con errores en la entrada de datos numéricos:

=====

Introduzca un número: gg

Incorrecto

Introduzca un número: 1234567

Introduzca el dígito a buscar (0 a 9): 99

No está entre 0 y 9

Introduzca el dígito a buscar (0 a 9): yyy

Incorrecto

Introduzca el dígito a buscar (0 a 9): 9

Posiciones en las que se encuentra el dígito 9, en el numero 1234567
 NO SE ENCUENTRA EL DÍGITO 9

=====