

## Primer problema (4 puntos)

La NFL nos ha encargado una clase para poder calcular el total de centímetros avanzados por los jugadores de un equipo de fútbol americano. En el fútbol americano se usan yardas (yards), pies (feet) y pulgadas (inches) para medir las longitudes.

- 1 yarda = 3 pies
- 1 pie = 12 pulgadas
- 1 pulgada = 2,54 centímetros

Para que no nos tengamos que preocuparnos demasiado por ello **nos dan la siguiente clase ya hecha**:

```
public class Distance {  
    // Construye una instancia de Distance a partir de la cadena yfi  
    // que tiene el formato: 15y12f7i  
    // (para el caso concreto de 15 yardas, 12 pies y 7 pulgadas)  
    // Las distancias son siempre positivas o cero.  
    public static Distance fromString(String yfi) { ... }  
  
    // Crea una instancia de Distance con valor 0y0f0i  
    public Distance() { ... }  
  
    // Añade la distancia other sobre el objeto receptor  
    public void add(Distance other) { ... }  
  
    // Devuelve el número total de pulgadas del objeto receptor  
    public int toInches() { ... }  
  
    // Devuelve una cadena correspondiente a la distancia que almacena  
    // el objeto receptor con el mismo formato usado en fromString.  
    public String toString() { ... }  
}
```

La clase que se ha de programar, llamada **DistanceAccumulator**, servirá para ir acumulando las distancias avanzadas por cada uno de los jugadores.

Tiene los siguientes métodos:

- **DistanceAccumulator(int numPlayers)**
  - Constructor que recibe un entero como parámetro que representa el número de jugadores.
  - **Podéis suponer que numPlayers es un número positivo..**
- **void hasAdvanced(int player, Distance distance)**
  - Acumula distance a la distancia que llevaba avanzada el jugador indicado por player.
  - Cada jugador se identifica por un número que va desde 1 hasta el valor de numPlayers.
  - Si el número de jugador no se corresponde con ningún jugador válido, esta operación no hace nada.
- **double totalCentimeters(int player)**
  - Devuelve el total de centímetros avanzados por el jugador dado.
  - Si el número no se corresponde con ningún jugador, se devuelve -1.0.

## Segundo problema (6 puntos)

Haced una clase que represente un programa principal que pida una cadena (String) al usuario formada por parejas de número de jugador#distancia, separadas por espacios, con el siguiente formato:

```
12#0y2f8i 20#5y0f7i 12#10y4f7i
```

**Podéis suponer que la línea no contiene ningún error de formato.**

El programa ha de escribir en la salida el total de centímetros avanzados por los jugadores que han avanzado algún centímetro. En este caso escribiría dos líneas de la forma:

```
12 - 1043,94
20 - 474,98
```

La NFL marca que los dorsales válidos son desde el 1 hasta el 99.

### Pistas:

- Descomponed la solución en funciones sencillas
- En el programa principal podéis hacer funciones que se pasen un objeto de la clase DistanceAccumulator como parámetro (si lo necesitan) o bien que ésta sea una variable de instancia de la clase del programa principal.

## Consideraciones generales

- Explicad brevemente lo que hacéis y por qué.
- No intentéis resolverlo todo de golpe: descomponed el problema en problemas más sencillos.
- Usad diagramas o lo que consideréis más adecuado para explicarlos.

## Métodos públicos que podéis necesitar

- `class ConsoleProgram`
  - o `String readLine(String message)`
  - o `void println(String str) / void println(int n) / void println(double d)`
  - o `void print(String str) / void print(int n) / void print(double d)`
- `class Integer`
  - o `static int parseInt(String str)`
- `class String:`
  - o `char charAt(int index)`
  - o `int indexOf(char c)`
  - o `int indexOf(String s)`
  - o `int length()`
  - o `String concat(String s)`
    - También podéis usar `+` para concatenar Strings
  - o `String substring(int p1, int p2)`
  - o `String substring(int p1)`
  - o `String trim()`
  - o `static String valueOf(int n)`
  - o `static String valueOf(double d)`
- `class StringTokenizer:`
  - o `new StringTokenizer(String str)`
  - o `new StringTokenizer(String str, String delims)`
  - o `new StringTokenizer(String str, String delims, boolean returnDelims)`
  - o `boolean hasMoreTokens()`
  - o `String nextToken()`