

## **ENTREGA E6 (0,5 puntos)**

Antes de empezar...

- Descargar el archivo **E6.zip** e importarlo a Eclipse.
- Renombrar el proyecto (tecla F2). Por ejemplo, para el grupo formado por Ana Pérez, Jon Azkue y Miren Landa el nuevo nombre sería: **E6\_APerezJAzkueMLanda**.
- En el fichero **componentesGrupo**, escribir los nombres de los componentes del grupo.

Además, tened en cuenta que:

- Se valorará la eficiencia de las soluciones, es decir, además de que sean correctas, deben ser eficientes.
- Aparte de los casos de prueba que se os entregan, se espera que incorporéis casos de prueba adicionales y significativos.
- Para entregarlo, debéis exportar el proyecto y subirlo a eGela.

**Fecha límite de entrega:** 17/12/2023 a las 23:59

—

### **Ayuda para importar:**

File -> Import... -> General -> Existing Projects into Workspace -> Select archive file (el .zip descargado) -> Finish

### **Ayuda para exportar:**

Pinchar en el proyecto. File -> Export... -> General -> Archive File -> (seleccionar las carpetas/archivos a exportar) -> To archive file (escribir una ruta y nombre para el nuevo archivo .zip) -> Finish

## **Ejercicio 1**

Un secuestrador quiere enviar una nota de rescate. Como todo buen secuestrador, en vez de escribir la nota a mano ha decidido usar palabras recortadas de una revista.

Debes completar el método *notaPosible*, que, dada la lista de palabras que quiere escribir en la nota y la lista de palabras recortadas de la revista, devuelve un booleano que indica si es posible escribir la nota completa usando palabras de la revista. Ten en cuenta que:

- No se debe distinguir entre minúsculas y mayúsculas.
- Se deben usar palabras de la revista tal y como están, es decir, no se pueden partir ni tampoco concatenar palabras para formar nuevas.
- El algoritmo debe ser lo más eficiente posible.

### ***Ejemplo 1:***

palabrasNota: give me one million TONIGHT

palabrasRevista: give me one million reasons to trust you tonight

Devolverá: true

### ***Ejemplo 2:***

palabrasNota: I need diamonds

palabrasRevista: you and I we are like diamonds in the sky

Devolverá: false (porque la palabra *need* no aparece en la revista)

### ***Ejemplo 3:***

palabrasNota: be quiet and be fast

palabrasRevista: washing machines should be fast and quiet

Devolverá: false (porque la palabra *be* solo aparece una vez en la revista)

## Ejercicio 2

Disponemos de una lista de médicos. Para cada médico conocemos su nombre, los nombres de los pacientes a los que tiene que atender durante el día de hoy, y la hora y minuto en los que dicho médico comenzará hoy las consultas:

<b>nom:</b>	<b>Dr. Arana</b>
<b>pacientes:</b>	<Ane, Jon, Lidia>
<b>hora:</b>	8
<b>minuto:</b>	45

<b>nom:</b>	<b>Dra. Loidi</b>
<b>pacientes:</b>	<Asier, Jon>
<b>hora:</b>	9
<b>minuto:</b>	0

<b>nom:</b>	<b>Dra. Urrutia</b>
<b>pacientes:</b>	<Jon, Oihana, Sara, Lidia>
<b>hora:</b>	9
<b>minuto:</b>	30

Se pide implementar el método **asignarCitas** de la clase **Ambulatorio**. Este método devuelve un **HashMap** (cuya definición debes completar) en el que para cada paciente, se guarda una lista con las citas que se han asignado a dicho paciente (nombre del médico, y hora:minuto de la cita). Al primero de los pacientes de cada médico se le asignará la hora y minuto en la que el médico comienza a pasar consulta, y al resto de los pacientes de dicho médico se les asignará una cita cada 15 minutos. No importa si al mismo paciente se le asignan dos citas a la misma hora con distinto médico. Además, suponemos que si aparece el mismo nombre de paciente para más de un médico, se trata del mismo paciente.

Puedes crear las clases auxiliares que consideres necesarias.

Para el ejemplo superior, el **HashMap** resultante debería contener lo siguiente (el orden de los pacientes no es significativo):

(clave)	(valor)
Oihana	[(Dra. Urrutia,9:45)]
Jon	[(Dr. Arana,9:0), (Dra. Loidi,9:15), (Dra. Urrutia,9:30)]
Ane	[(Dr. Arana,8:45)]
Sara	[(Dra. Urrutia,10:0)]
Asier	[(Dra. Loidi,9:0)]
Lidia	[(Dr. Arana,9:15), (Dra. Urrutia,10:15)]