

FUNDAMENTOS DE PROGRAMACIÓN EXAMEN PRÁCTICO 2

(Enero 2016)

!Importante! No se corregirá ninguna práctica que no cumpla escrupulosamente las normas que aparecen a continuación

-
- Para iniciar sesión en las aulas de prácticas, tendrá que introducir su identificador de usuario y contraseña como siempre. En la casilla correspondiente a **Código: examenfp**. Con este código sólo tendrá acceso Internet a la página de la asignatura. No tendrá disponible su unidad en red U: ni los puertos USB.
 - Se creará un fichero fuente con la solución del problema, nombrándolo **examen.cpp**.
 - Debe contener en la cabecera, `//NOMBRE: nombre,apellidos, DNI`
`//PROBLEMA: conjunto`
 - En la evaluación de las prácticas se tendrá en cuenta, además de la corrección de la solución propuesta, la modularización de la clase, el estilo de programación, el uso correcto de espacios y tabuladores, así como la claridad del código fuente.
 - La entrega de la práctica se hará durante el tiempo de duración del examen, y se hará en entrega de prácticas accesible a partir de la página de la asignatura `decsai.ugr.es` (a través de Entrega prácticas). Bien directamente `https://150.214.191.180` en caso de problemas .
 - La práctica se puede entregar tantas veces como se quiera durante el examen. El sistema se queda con la última entrega. De hecho, **se recomienda que se entregue varias veces a lo largo del examen**, ya que si el ordenador se quedara colgado, habría que reiniciarlo y se perdería toda la información.
 - Tiempo de examen: **1.30 HORA**
-

Enunciado del problema

Se quiere trabajar con una secuencia de enteros dada en la entrada. Apartir de la secuencia anterior se pide obtener otra secuencia de enteros que denominaremos *simplificada*, que contiene los valores diferentes hallados en la secuencia original (esto es, sin valores repetidos). Antes de finalizar se mostrará el contenido de la secuencia *simplificada* desde el programa principal.

La secuencia de enteros dada en la entrada es de longitud variable, y esta longitud se desconoce apriori, por ello la entrada de datos termina cuando se introduce 0, (no se considera como valor propio dentro de la secuencia).

Para la implementación del ejercicio, previamente será necesario:

Definir la clase *SecuenciaEnteros* con los datos miembros, el (los) constructor(es) que se vayan a utilizar, y los métodos auxiliares de *TotalUtilizados* para obtener el número de componentes utilizados, *Capacidad* para conocer el número de enteros máximo que puede albergar, *Aniade* para añadir un componente, *Elemento* para obtener un componente ubicado en una determinada posición y *Buscar* para que devuelve la posición de un determinado valor en caso de encontrarse, o bien -1 en caso contrario. Adicionalmente se puede implementar cualquier otro método que considere oportuno para la realización del ejercicio.

Por último, para la solución del ejercicio se pide implementar un método llamado **Simplifica** que devuelve la secuencia resultante sin elementos repetidos.

Nota:

se tendrá en cuenta el grado de eficiencia en su implementación.

Por ejemplo, para el siguiente secuencia de datos de entra:

3 3 1 7 1 9 1 3 0

el contenido de la secuencia *simplificada* es:

3 1 7 9