Universidad de Granada
Departamento de Ciencias de la Computación
e Inteligencia Artificial

Fundamentos de Programación (2014/15)
1º Grado en Ingeniería Informática (Teoría)
10 de Septiembre de 2015

DECSAI

Normas para la realización del examen: **Duración: 2 horas**

- **Debe disponer de un documento oficial que acredite su identidad a disposición del profesor.**
- **Escriba su nombre completo, DNI y grupo en todos y cada uno de los folios que entregue.**

◁ Ejercicio 1 ▷ Números amigos [2 puntos]

Dos números *amigos* son dos números naturales a y b, tales que la suma de los divisores propios de a más uno es igual a b, y viceversa.

Un ejemplo de números amigos es el par de naturales $(220, 284)$, ya que:

- Los divisores propios de 220 son 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, que suman 283, y $283 + 1 = 284$.
- Los divisores propios de 284 son 2, 4, 71 y 142, que suman 219, y $219 + 1 = 220$.

Realizar un programa que realice estas dos tareas:

1. En primer lugar debe leer dos números naturales e indicar si son o no amigos
2. A continuación leerá otro número natural, n, e informará si existe algún número amigo de n en el intervalo centrado en n y de radio $n/3$.

◁ Ejercicio 2 ▷ Métodos sobre `SecuenciaEnteros` [2 puntos]

Defina la clase `SecuenciaEnteros` para representar una secuencia de enteros. Debe contener métodos para añadir al final un nuevo entero, comprobar cuántos hay y recuperar el valor que haya en un índice dado. Para almacenar internamente los enteros, use un *array* de enteros.

Defina métodos para resolver las siguientes tareas:

1. Defina dentro de la clase lo que sea necesario para sumar dos secuencias.
   - Las dos secuencias han de tener el mismo número de componentes.
   - La suma de dos secuencias es otra secuencia resultante de sumar las correspondientes componentes.

   Por ejemplo, la suma de `{2,5,7}` y `{4,2,1}` es `{6,7,8}`.
2. Defina dentro de la clase lo que sea necesario para calcular el número de cadenas consecutivas ascendentes de números.
   Por ejemplo, la secuencia `{2,4,1,1,7,2,1}` tiene 4 cadenas ascendentes que son `{2,4}`, `{1,1,7}`, `{2}` y `{1}`.

◁ Ejercicio 3 ▷ Buscaminas [3 puntos]

Buscaminas es un juego muy conocido cuyo objetivo es encontrar todas las minas existentes en un tablero rectangular, sin abrir ninguna. Si se abre una mina, perderá la partida.

Se trabajará con una tabla de datos $n \times m$ en la que todas las filas tienen el mismo número de columnas y los datos son de tipo `bool`. La clase se llamará `TableroBuscaMinas` y contendrá valores `true` en caso de haber una mina en la casilla especificada y `false` en caso contrario.

1. Defina los datos miembros de la clase `TableroBuscaMinas`.
2. Implemente un constructor que reciba el número de filas y columnas que tendrá el tablero y lo inicie fijando las casillas a `false` (*no hay mina*).
   Nota: El número de filas y columnas disponibles para construir tableros puede ser mayor que las que realmente se van a utilizar.
3. Escriba un método para modificar el estado de una casilla del tablero.
4. Implemente un método que reciba las coordenadas de una casilla y devuelva un valor entero que indique el número de minas que rodean a la misma, incluyendo las diagonales.
   - En caso de que la casilla no contenga una mina, el valor devuelto estará comprendido en $[0, 8]$.
   - En caso de que la casilla contenga una mina, se devolverá el valor -1.

   Nota: Hay que tener en cuenta que las casillas de los bordes deben tratarse de manera diferenciada.

5. Implemente un método que devuelva **otra** tabla de datos de igual tamaño al tablero y que contenga valores de tipo `int` con la solución.
   - La solución contiene en cada posición un valor que indica el número de minas adyacentes, y un valor de -1 en aquellas donde está situada una mina.
     Por ejemplo, dado el tablero de la izquierda (aparecen siete minas) el resultado sería el mostrado en la tabla de la derecha.

			•
	•		
			•
	•	•	•
		•	

1	1	2	-1
1	-1	3	2
2	3	5	-1
1	-1	-1	-1
1	3	-1	3
0	1	1	1

   - La tabla de datos original **no** se modifica.

   Nota: Suponga que existe la clase `TablaRectangularEnteros`.