

# Grado en Ingeniería Informática y Grado en Estadística Fundamentos de Programación

Examen convocatoria ordinaria. 11 de enero de 2018.

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_ Grupo \_\_\_\_\_

--	--	--	--	--	--

DNI y Firma \_\_\_\_\_

Duración del examen: 3 horas.

Empezar cada problema en una cara distinta.

Poner nombre y apellidos en todas las páginas.

Las hojas del enunciado también deben entregarse.

Se valorará la presentación y la claridad en la exposición.

Se valorará la adecuación de las estructuras utilizadas al problema a resolver.

No se calificarán las respuestas escritas a lápiz.

1. Cuando se representan valores discretos (enteros) en un plano, dos puntos son contiguos cuando de sus dos coordenadas una es idéntica y la otra consecutiva. Por ejemplo, los puntos (3, 2) y (3, 3) son consecutivos pero (3, 2) y (5, 2) no lo son. Suponiendo que para representar los puntos del plano se utilizan objetos de tipo **Point**:
  - a) [1 pto] Elaborar un método Java que determine si dos puntos son contiguos o no.

```
public static boolean contiguos (Point p1, Point p2){
    boolean contig = false;
    if (p1.x == p2.x)
        {if ((p1.y - p2.y) == 1) || ((p1.y - p2.y) == -1))
            contig = true;}
    else
        if (p1.y == p2.y)
            if ((p1.x - p2.x) == 1) || ((p1.x - p2.x) == -1))
                contig = true;
    return contig;
}
```

- b) [1,5 ptos] Implementar un método Java que, dado un **array** de **Point**, determine si los puntos que representan forman o no una cadena. Se dice que varios puntos forman una cadena cuando cada uno de ellos es contiguo al siguiente. Es decir, los puntos a, b y c forman una cadena si a y b son contiguos y b y c también lo son.

```
public static boolean cadena (Point [] v){
    boolean cad = true;
    int i = 0;
    while ((i < v.length-1) && cad){
        cad = contiguos (v[i], v[i+1]);
        i++;
    }
    return cad;
}
```

- c) [0,5 ptos] Definir las estructuras de datos Java necesarias para poder almacenar la información del array del apartado anterior en una lista enlazada.

```
public class Nodo {  
    Nodo sgte;  
    Point dato;  
    public Nodo (Point p){  
        this.sgte = null;  
        this.dato = new Point (p.x, p.y);  
    }  
}
```

```
public class Lista {  
    Nodo primero;  
    public Lista (){  
        this.primeros = null;  
    }  
}
```

2. [1,5 ptos] Dado el fichero de texto **entrada.txt** que contiene números enteros, crear un programa Java que escriba en el fichero de texto **salida.txt** dos mensajes, cada uno en una línea. El primer mensaje debe indicar cuántos números hay en el fichero de entrada y el segundo cuántos son pares. Supóngase que el fichero de entrada existe y que están importados los paquetes necesarios.

**RECORDATORIO:** Para declarar y abrir en modo escritura un fichero de texto:

```
PrintWriter <id_fich> = new PrintWriter (new FileWriter <nombre_fich>);
```

Al abrir el fichero se puede producir la excepción `IOException`.

Para declarar y abrir en modo lectura un fichero de texto:

```
Scanner <id_fich> = new Scanner (new File(<nombre_fich>));
```

Al abrir el fichero se puede producir la excepción `FileNotFoundException`.

```
import java.io.*;
import java.util.Scanner;
public class CuentaFichero{
    public static void main (String[] args){
        Scanner ent;
        PrintWriter sal;
        try {
            ent = new Scanner (new File ("entrada.txt"));
        }
        catch (FileNotFoundException e) {
            System.out.println ("No se encontró el fichero");
            return;
        }
        try {
            sal = new PrintWriter ("salida.txt");
        }
        catch (IOException e) {
            System.out.println ("No puede abrirse el fichero");
            return;
        }
        int num;
        int cont = 0;
        int contPares = 0;
        while (ent.hasNextInt()){
            num = ent.nextInt();
            cont++;
            if (num % 2 == 0)
                contPares++;
        }
        sal.println ("La catidad de numeros es " + cont);
        sal.println ("La catidad de numeros pares es " + contPares);
        ent.close();
        sal.close();
    }
}
```

3. [1 pto] Crear una función Java **recursiva** que multiplique dos números naturales mayores que cero a partir de la siguiente definición de producto:

$$a * b := \begin{cases} a & \text{si } b = 0 \\ a * (b - 1) + a & \text{si } b > 0 \end{cases}$$

```
public static int producto (int a, int b){  
    // a>0, b>0  
  
    if (b == 0)  
        return a;  
    else  
        return (producto (a, b-1) + a);  
}
```



4. Para analizar una imagen de una porción de cielo se obtiene una matriz de 40x20 componentes enteras, cada una de las cuales indica la intensidad de luz recibida en el punto (la intensidad es un valor entre 0 y 100).

Se considera que hay una estrella en una componente de la matriz, cuando la suma de las intensidades correspondientes a esa componente más la media de los 8 puntos que la circundan es mayor o igual que 60. Se supone que en el borde de la matriz no existen estrellas.

Para representar la imagen de estrellas correspondiente se utiliza una matriz de caracteres, en la que las posiciones correspondientes a estrellas contienen el carácter \* y el resto, un espacio en blanco.

- a) [1,5 ptos] Elaborar un método Java que tome como entrada una matriz de intensidades y devuelva una matriz con la imagen de estrellas resultante.

```
public static char [][] imagen (int [][] inten){
    int i, j, sum = 0;
    double media;
    char [] imag = new char [inten.length][inten[0].length];

    for (i=0; i<imag.length; i++)
        for (j=0; j<imag[0].length; j++)
            imag[i][j] = ' ';

    for (i=1; i<inten.length-1; i++){
        for (j=1; j<inten[0].length-1; j++){
            sum = inten[i-1][j-1] + inten[i-1][j] + inten[i-1][j+1]
                + inten[i][j-1] + inten[i][j+1] + inten[i+1][j-1]
                + inten[i+1][j+1];
            media = sum/8.0;
            if (media+inten[i][j] >= 60)
                imag [i][j] = '*';
        }
    }
    return imag;
}
```

- b) [0,5 ptos] Elaborar un método Java que tome como entrada la matriz con la imagen de estrellas y devuelva el número de estrellas que contiene esa imagen.

```
public static int estrellas (char [][] img){
    int cont = 0;
    for (int i=1; i<img.length-1; i++)
        for (int j; j<img[0].length-1; j++)
            if (img[i][j] == '*')
                cont++;
    return cont;
}
```

- c) [1 pto] Definir la estructura de datos completa que permita almacenar la matriz de intensidades y el número de estrellas que contiene.

```
public class Reg {
    int [][] inten;
    int est;
    public Reg () {
        this.inten = new int [40][20];
        this.est = 0;
    }
    public Reg (int e, int[][]m) {
        this.est = e;
        this.inten = new int [m.length][m[0].length];
        for (int i=0; i<m.length; i++)
            for (int j; j<m[0].length; j++)
                this.inten [i][j] = m[i][j];
    }
}
```

5. [1,5 ptos] Elaborar un programa Java que lea líneas de caracteres de teclado hasta que el usuario introduzca la cadena vacía. Por cada línea de entrada que se lea, el programa debe escribir en pantalla el porcentaje de letras que hay en esa cadena. Cuando se hayan leído todas las líneas de entrada el programa debe escribir en pantalla el porcentaje de letras que han leído en total. Supóngase que el idioma utilizado es el inglés (no hay, ni tildes ni diéresis).

```
import java.util.Scanner;

public class Porcentaje{

    public static void main (String[] args){
        Scanner in = new Scanner (System.in);
        char c;
        int i, contLetras;
        int contLetrasTotal = 0;
        int contCar = 0;

        System.out.println ("Teclee una cadena de caracteres: ");
        String linea = in.nextLine();

        while (linea.length > 0){
            contCar = contCar + linea.length;
            contLetras = 0;
            for (i=0; i<linea.length; i++){
                c = linea.charAt(i);
                if (((c>'a')&&(c<'z')) || ((c>'A')&&(c<'Z'))){
                    contLetras ++;
                }
            }
            contLetrasTotal = contLetrasTotal + contLetras;
            System.out.println ("Hay "+(contLetras*100/contCar)+" de letras");
            System.out.println ("Teclee otra cadena de caracteres: ");
            linea = in.nextLine();
        }

        System.out.println ();
        System.out.println ("Hay "+(contLetrasTotal*100/contCar)+" de letras");
    }
}
```