

Repetir

- Leer la linea

- Sacar el carácter iésimo de esa linea

- Comprobar que es válido

* ~~Comprobar~~ Convertirlo al siguiente

Concatenarlo a una nueva cadena

-

```
String clave = "";
```

char c;

```
String linea;
```

~~int cout = 0;~~

```
while (f.hasNextLine()) {
```

linea = f.nextLine();

~~if (linea.length > cout)~~

= linea.charAt(cout);

if ((c > 'a') & (c < 'z')) || ((c > 'A') & (c < 'Z'))

clave = clave + convertir(c);

else

~~if (c == ' ')~~

~~clave = clave~~

if (c == ' ')

clave = clave + c;

else

System.out.println("Clave erronea");

~~cout++;~~

cout++

{ else

System.out.println("Clave erronea");

}

$\in [a, z]$

$[A, Z]$

public static char convertor ($\leftarrow \text{char } c\right)$ {

if ($c == 'a'$)
return 'z';

else if ($c == 'A'$)
return 'Z';

else return (char) ((int) $c - 1$),

}

- Main.java

```
public class Main1 {
    public static void main(String[] args) {
        //Ejercicio 1
        final int NUM = 9;
        TPersona [] persona = new TPersona[NUM];
        crearRed (persona);
        mostrarRed (persona);
        System.out.println ("La edad mÃ¡s frecuente es " +
edadMasFrecuente (persona));
    }

    public static int edadMasFrecuente(TPersona[] p) {
        int edad = p[0].edad, edadMax = edad;
        int frecuencia = 1, freqMax = 1;
        for (int i = 1; i<p.length; i++) {
            if (edad == p[i].edad) {
                frecuencia++;
            } else { // cambio de edad
                if (frecuencia > freqMax) {
                    freqMax = frecuencia; edadMax = edad;
                }
                edad = p[i].edad; frecuencia = 1;
            }
            if (frecuencia > freqMax) {
                freqMax = frecuencia; edadMax = edad;
            }
        }
        return edadMax;
    }

    public static void mostrarRed(TPersona[] p) {
        for (int i = 0; i<p.length; i++)
            System.out.println (p[i].edad + " " + p[i].nombre + " "
+ p[i].apellidos);
    }

    public static void crearRed(TPersona[] persona) {
        persona[0] = new TPersona (19, "Angel", "SÃ¡nchez PÃ©rez");
        persona[1] = new TPersona (19, "VÃCTOR", "Marcos GonzÃ¡lez");
// 19/20 ¿cuÃ¡l serÃ¡ la salida?
        persona[2] = new TPersona (20, "Carlos", "GÃ³mez Alonso");
        persona[3] = new TPersona (20, "Javier", "PÃ©rez Prieto");
        persona[4] = new TPersona (21, "AdriÃ¡n", "RodrÃ�guez SÃ¡nchez");
        persona[5] = new TPersona (21, "Laura", "PÃ©rez GonzÃ¡lez");
        persona[6] = new TPersona (21, "Silvia", "Alonso GÃ³mez"); //21
/ 22 ¿funcionarÃ¡ bien?
```

```

    persona[7] = new TPersona (22, "Laura", "Marcos García");
    persona[8] = new TPersona (22, "Pedro", "Álvarez Roldán");
}

}

class TPersona {
    int edad;
    String nombre, apellidos;
    public TPersona(int edad, String nombre, String apellidos) {
        this.edad = edad;
        this.nombre = nombre;
        this.apellidos = apellidos;
    }
}

```

-Main2.java

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Main2 {
    public static void main(String[] args) {
        // Ejercicio 2
        try {
            Scanner f = new Scanner (new File ("fclave.txt"));
            String clave = "";
            boolean error = false;
            int indice = 0;
            while (f.hasNextLine() && ! error) {
                String linea = f.nextLine(); int l =
linea.length();
                if (indice < l){
                    char c = linea.charAt(indice);
                    if (c==' ') clave += c;
                    else if (esLetra (c)) clave +=
cambiada(c);
                    else error = true;
                } else {
                    error = true;
                }
                indice++;
            }
            if (error)
                System.out.println ("Clave errónea");
            else
                System.out.println ("La clave es " +
clave);
            f.close();
        }
        catch (FileNotFoundException e) {
            System.out.println ("Fichero erróneo");
        }
    }

    public static char cambiada(char c) {
        // PRE : c es una letra
    }
}

```

```

    if (c=='a')
    else if (c == 'A')      return 'Z';
    else                      return (char)(c-1);
}

public static boolean esLetra(char c) {
    return (c<='Z' && c>='A' || c>='a' && c<='z');
}

~ Main3.java

import java.util.Scanner;
public class Main3 {

    public static void main(String[] args) {
        // Ejercicio 3
        int[][] m = {
            { 1, 2, 3, 4 },
            { 11, 12, 13, 14 },
            { 21, 22, 23, 24 },
            { 31, 32, 33, 34 },
            { 41, 42, 43, 44 },
            { 51, 52, 53, 54 }
        };
        Scanner in = new Scanner(System.in);
        imprimir(m);
        int rot = in.nextInt();
        while (rot >= 0) {
            rotar(m, rot);
            System.out.println("rotada " + rot);
            imprimir(m);
            rot = in.nextInt();
        }
    }

    public static void rotar(int[][] m, int num) {
        // PRE:
        // OBS: Si num <=0, no se rota
        // Hacia arriba
        // Haciendo num veces una rotaciÃ³n
        for (int rot = 1; rot <= num; rot++) {
            // una rotaciÃ³n hacia arriba
            for (int col = 0; col < m[0].length; col++) {
                int aux = m[0][col];
                for (int i = 0; i < m.length - 1; i++) {
                    m[i][col] = m[i+1][col];
                }
                m[m.length - 1][col] = aux;
            }
        }
    }

    public static void rotarb(int[][] m, int num) {
}

```

```

    //hacia arriba (mÁs eficiente), no va una a una sino de golpe
    ...
    for (int rot = 1; rot <= num; rot++) {
        // una rotaciÁn hacia arriba
        for (int col = 0; col < m[0].length; col++) {
            int aux = m[0][col];
            for (int i = 0; i < m.length - 1; i++) {
                m[i][col] = m[i + 1][col];
            }
            m[m.length - 1][col] = aux;
        }
    }
}

private static void imprimir(int[][] m) {
    for (int i = 0; i < m.length; i++) {
        for (int j = 0; j < m[0].length; j++) {
            System.out.printf ("%3d ", m[i][j]);
        }
        System.out.println ();
    }
}
}

```

- Main3b.java -

```

import java.util.Scanner;

public class Main3b {

    public static void main(String[] args) {
        // Ejercicio 3
        int[][] m = {
            { 1, 2, 3, 4 },
            { 11, 12, 13, 14 },
            { 21, 22, 23, 24 },
            { 31, 32, 33, 34 },
            { 41, 42, 43, 44 },
            { 51, 52, 53, 54 }
        };
        Scanner in = new Scanner(System.in);
        imprimir(m);
        int rot = in.nextInt();
        while (rot >= 0) {
            rotar(m, rot);
            System.out.println("rotada " + rot);
            imprimir(m);
            rot = in.nextInt();
        }
    }

    public static void rotar(int[][] m, int num) {
        // PRE: num >= 0
        // Hacia arriba, num posiciones
        // Usando otra matriz
        // En cada posiciÁn se pone el que hay num posiciones mÁs
        // abajo
        // pero si nos pasamos, volvemos a contar desde el principio:
    }
}

```

```

// 1Aº, el 9Aº el 2Aº ... // Si hay / filas, el 7Aº es el 0, el 8Aº es el
// resto de la división entera!
// AdemÁs, si se piden tantas rotaciones que el nÁmero de
columnas, es como no rotar ...
// y si nos piden mÁs... vuelve a ser el resto de la
divisiÁn entera!

int nfilas = m.length, ncolumnas = m[0].length;
int [][]nueva = new int [nfilas][ncolumnas];

num = num % nfilas; // reducir el nÁmero de rotaciones
// crear nueva matriz, con la rotada
for (int i = 0; i<nfilas; i++)
    for (int j= 0; j<ncolumnas; j++){
        nueva [i][j] = m[ (i+num)%nfilas ][j];
    }
// copiarla en la original
for (int i = 0; i<nfilas; i++)
    for (int j= 0; j<ncolumnas; j++){
        m [i][j] = nueva [i][j];
    }
}

private static void imprimir(int[][] m) {
    for (int i = 0; i<m.length; i++){
        for (int j = 0; j<m[0].length; j++)
            System.out.printf ("%3d ", m [i][j]);
        System.out.println ();
    }
}
}

```

—Main3c.java—

```

import java.util.Scanner;

public class Main3c {

    public static void main(String[] args) {
        // Ejercicio 3
        int [][] m = {
            { 1, 2, 3, 4 },
            { 11, 12, 13, 14 },
            { 21, 22, 23, 24 },
            { 31, 32, 33, 34 },
            { 41, 42, 43, 44 },
            { 51, 52, 53, 54 },
            { 61, 62, 63, 64 },
            { 71, 72, 73, 74 }
        ,
            { 81, 82, 83, 84 },
            { 91, 92, 93, 94 }
        };
        Scanner in = new Scanner(System.in);
        imprimir(m);
        int rot = in.nextInt();
    }
}

```

```

        while (rot >= 0) {
            rotar(m, rot);
            System.out.println("rotada " + rot);
            imprimir(m);
            rot = in.nextInt();
        }

    }

public static void rotar(int[][] m, int k) {
    //PRE : k>=0
    // hacia arriba
    // cada elemento se sustituye por el que estÃ¡i k posiciones
    // mÃ¡s adelante, mÃ³dulo el tamaÃ±o de la columna
    // esto genera cadenas (grupos).
    // El elemento de la posiciÃ³n i debe ser sustituido por
    // el de la posiciÃ³n i+k,
    //          Ã©ste por el de la posiciÃ³n i+k,
    //          Ã©ste por el de la posiciÃ³n i+k de nuevo , etc
    //          hasta volver a encontrar la posiciÃ³n de inicio
    // de grupo.
    //          El primer grupo empieza en 0, el segundo en 1
    ...
    //          Por ejemplo para el caso de 6 filas y 3
    // rotaciones los grupos son
    //          0 3 (0), 1 4 (1) y 2 5 (2)
    // para 2 rotaciones los grupos son
    //          0 2 4 (0), 1 3 5 (1)
    // y para 5 rotaciones, el grupo es
    //          0 5 4 3 2 1 (0)
    // para el caso de 10 filas y 6 rotaciones, los
    // grupos son
    //          0 6 2 8 4 (0) , 1 7 3 9 5 (1)
    int numfilas = m.length, numcolumnas = m[0].length;
    k = k % numfilas; // en un ejemplo de 6 filas. 6 rotaciones es
    lo mismo que ninguna, 7 es lo mismo que 1 ...
    if (k > 0) {
        // k rotaciones hacia arriba
        int numgrup = MCD(numfilas, k); // k<numfilas
        System.out.println(numgrup + " grupos");
        for (int grup = 0; grup < numgrup; grup++) {
            for (int col = 0; col < m[0].length; col++) {
                int i = grup;
                System.out.print("grupo " + i + " : ");
                int aux = m[grup][col];
                int j = (i + k) % numfilas; // k
                posiciones mÃ¡s adelante
                while (j != grup) {
                    System.out.print(i + " - " + j
                    + ";   ");
                    m[i][col] = m[j][col];
                    i = j;
                    j = (i + k) % numfilas;
                }
                m[i][col] = aux;
                System.out.println(" y " + i + " - " +
grup);
            }
        }
    }
}

```

```

private static int MCD(int m, int n) {
    // m, n>0
    if (m < n) {
        int aux = m;
        m = n;
        n = aux;
    }
    int r = m % n;
    while (r > 0) {
        m = n;
        n = r;
        r = m % n;
    }
    return n;
}

private static void imprimir(int[][] m) {
    for (int i = 0; i < m.length; i++) {
        for (int j = 0; j < m[0].length; j++)
            System.out.printf("%3d ", m[i][j]);
        System.out.println();
    }
}

```

Main4.java

```

import java.util.Scanner;

public class Main4 {

    public static void main(String[] args) {
        // Ejercicio 4
        Scanner in = new Scanner (System.in);
        int n = in.nextInt();
        escribeBinario0(n);
    }

    private static void escribeBinario0(int n) {
        // escribeEl Binario, usando un recursivo
        // PRE: n>=0
        // Si no, escribe 0
        if (n>0) escribeBinario(n);
        else System.out.print (0);
    }

    private static void escribeBinario(int n) {
        // escribeEl Binario, recursivo
        // PRE: n>0
        if (n>0) {
            escribeBinario (n/2);
            System.out.print (n%2);
        }
    }
}

```

Main5.java

```
import java.util.Scanner;
public class Main5 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Nodo p, q, r;
        p = new Nodo(1);
        r = p;
        r.sig = new Nodo(2);
        r = r.sig;
        r.sig = new Nodo(3);
        r = r.sig;
        r.sig = new Nodo(4);
        p.imprime();

        System.out.println("\n-----");

        // r = null;
        // System.out.println (r);
        // System.out.println (r==null);
        // r.imprime2();

        // r = new Nodo(100);
        // System.out.println (r);
        // System.out.println (r==null);
        // r.imprime2();

        // System.out.println (r.sig);
        // System.out.println (r.sig==null);

        q = p;
        q = q.sig;
        p.sig = q.sig;
        q.sig.sig.sig = q.sig;
        p.imprime();
        System.out.println ();
        q.imprime();
    }
}

class Nodo {
    int dato;
    Nodo sig;

    public Nodo(int dato) {
        this.dato = dato;
        this.sig = null;
    }

    public void imprime() {
        Scanner entrada = new Scanner(System.in);
```

```

        if (this != null) {
            System.out.print(this.dato);
            if (this.sig != null) {
                System.out.print(" (" + this.sig + "), ");
                System.out.print("MÃ;s: 1; Terminar: -1 ");
                int seguir = entrada.nextInt();
                if (seguir == -1) return;
                this.sig.imprime();
            } else {
                System.out.println(".");
            }
        }
    public void imprime2() {

        if (this != null) {
            System.out.print(this.dato);
            System.out.print(" (" + this.sig + "), ");
            this.sig.imprime();
        } else { // Â;DEAD CODE!
            System.out.println(".");
        }
    }
}

```

Adjuntos:

Main1.java	1,8 KB
Main2.java	1,1 KB
Main3.java	1,5 KB
Main3b.java	1,7 KB
Main3c.java	2,7 KB
Main4.java	591 bytes
Main5.java	1,5 KB

Grado en Ingeniería Informática y Grado en Estadística

Fundamentos de Programación

Examen convocatoria extraordinaria. 9 de febrero de 2016

Apellidos _____

Nombre _____

Grupo _____

DNI y Firma _____

Duración del examen: 2 horas y media.

Resolver cada problema en una página diferente.

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. Una red social tiene almacenados datos de sus afiliados en un vector de registros de tipo **TPersona**, ordenados por el campo **edad**. Se desea procesar este vector para conocer la edad más común (la que aparece un mayor número de veces) entre ellos. A continuación se muestra un ejemplo del contenido del vector.

	Edad	Nombre	Apellidos
0	19	Angel	Sánchez Pérez
1	19	Víctor	Marcos González
2	20	Carlos	Gómez Alonso
3	20	Javier	Pérez Prieto
4	21	Adrián	Rodríguez Sánchez
5	21	Laura	Pérez González
6	21	Silvia	Alonso Gómez
7	35	Laura	Marcos García
8	42	Pedro	Álvarez Roldán

- [0,5 ptos] Definir la clase Java (con un constructor) que permita representar un registro **TPersona**.
- [2 ptos] Elaborar un método Java tome como entrada el vector a procesar y devuelva la edad más frecuente (si hay más de una edad más frecuente que muestre una cualquiera). En el ejemplo anterior el método devolvería 21. **El vector debe recorrerse una sola vez.**
- [0,25 ptos] ¿Qué valor devolvería tu método del apartado anterior si en el ejemplo Víctor tuviese 20 años? Detalla brevemente como modificarías tu método para que devolviese el otro valor (20 ó 21).

2. [3 ptos] Se tiene un fichero de texto, **fclave.txt**, que contiene una clave oculta. El método para obtener la clave consiste en:

- Se toma el primer carácter de la primera línea, el segundo de la segunda, el tercero de la tercera, etc., formándose con ellos una cadena de caracteres.
- En la cadena obtenida se sustituye cada carácter por el anterior en la tabla de caracteres (**b** se sustituye por **a, c** por **b, d** por **c, a** por **z** etc). Los espacios no se modifican. La cadena resultante es la clave buscada.

Por ejemplo, la clave escondida en el siguiente fichero es **Cantor**:

De Morgan
 Abel
 Boole
 Brouwer
 Sierpinski
 Weierstrass

Escribir un programa en Java que obtenga la clave escondida en el fichero y la muestre por pantalla si no detecta ningún error, en caso contrario deberá mostrar el mensaje "Clave errónea". Los errores que se pueden producir al construir la clave son los siguientes:

- La n -ésima línea tiene una longitud menor de n , por lo que no es posible tomar su n -ésimo carácter.
- Algun carácter de la clave obtenida no es un espacio o una letra (A..Z, a..z).

RECORDATORIO: 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`.

3. [2,5 ptos] Escribir un método Java que tome como entrada un número entero y rote hacia arriba las filas de la matriz tantas veces como indique el número. Por ejemplo, si el número de entrada fuese 2 y la matriz de entrada la de la izquierda, el resultado debería ser la matriz de la derecha.

10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

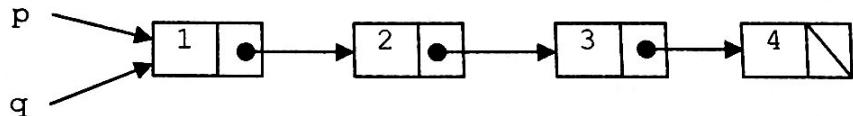
30	31	32	33	34
40	41	42	43	44
10	11	12	13	14
20	21	22	23	24

4. [1 pto] Codificar un método Java recursivo que tome como entrada un número entero estrictamente positivo y escriba en pantalla la secuencia de caracteres ceros y unos que representan ese número en binario.

INDICACIÓN: La secuencia binaria correspondiente al número 24, es equivalente a la secuencia binaria del número 12 seguida de un cero; la del 25 es la del 12 seguida de un uno.

5. [0,75 ptos] Dadas las variables `p` y `q` de tipo `Nodo`, suponiendo que en un momento dado de la ejecución se tiene el esquema de la derecha:

```
public class Nodo{  
    int dato;  
    Nodo sig;  
    ...  
}
```



A partir de esta situación, dibujar (en esta misma página) el esquema que se obtendría tras la ejecución del siguiente fragmento de código Java.

```
{  
    q = q.sig;  
    p.sig = q.sig;  
    q.sig.sig.sig = q.sig;  
}
```