

Programación II - 2010/2011 - Universitat Jaume I

Evaluación continua - Módulo 0 de teoría - Grupo A

22 de febrero de 2011

La duración máxima de esta prueba es de 45 minutos. No puedes consultar libros ni apuntes.

Ejercicio 1 (4 puntos, 1 punto por apartado)

Indica lo que se escribe en la salida estándar al ejecutar cada uno de los siguientes programas. No es necesario que expliques por qué.

```
(a) public class EjercicioA {
    public static void main (String[] args) {
        int a = 2;
        System.out.println("resultado " + (a += 3) + (a *= 4) );
        System.out.println("a " + a);
    }
}
```

```
(b) public class EjercicioB {
    public static void main (String[] args) {
        int a = 4;
        int b = a++ + a++ * ++a;
        System.out.println("a " + a);
        System.out.println("b " + b);
    }
}
```

```
(c) import java.util.*;
public class EjercicioC {
    public static void mitad (int numero) {
        numero /= 2;
        System.out.println("numero " + numero);
    }
    public static void mitad (int [] vector) {
        for (int i = 0; i < vector.length; i++) {
            vector[i] /= 2;
            System.out.println("vector " + vector[i]);
        }
    }
    public static void main (String[] args) {
        int a = 9;
        int b = a;
        int [] v = {5, 10};
        int [] w = v;
        mitad(a);
        mitad(v);
        System.out.println("a " + a);
        System.out.println("b " + b);
        System.out.println ("v " + Arrays.toString(v));
        System.out.println ("w " + Arrays.toString(w));
    }
}
```

```
(d) public class EjercicioD {
    public static void main (String[] args) {
        int a = 1, b = a;
        if ( (a = 3) == 3 || ! ( (b = 5) != 5 ) )
            System.out.println("Hola.");
        System.out.println("a " + a);
        System.out.println("b " + b);
    }
}
```

Ejercicio 2 (6 puntos)

Escribe un método `leerPalabrasDiferentes(n)` que reciba como argumento un entero n y lea palabras de la entrada estándar hasta tener n palabras diferentes. El método debe devolver como resultado un vector de talla n en el que cada componente sea una de las n palabras diferentes obtenidas. Esas palabras deben aparecer en el vector en el mismo orden en que han aparecido en la entrada.

Para resolver el problema, puede resultarte útil escribir y utilizar un método auxiliar que compruebe si una palabra ya está en el vector (aunque no es imprescindible que lo hagas así).