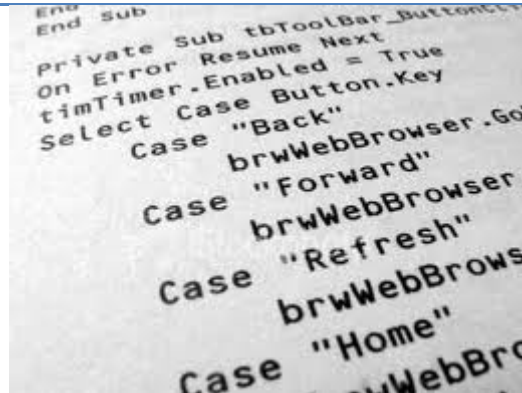
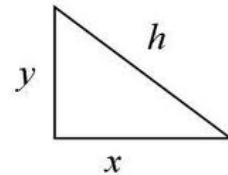


# Evaluación 1er Trimestre (Resuelto)



**Ejercicio 1** (2 puntos): Realizar un programa en PSEUDOCÓDIGO que dados dos CATETOS ( $x$  e  $y$ ), nos calcule la HIPOTENUSA ( $h$ ) según la fórmula  $h^2 = x^2 + y^2$ . Los valores  $x$  e  $y$  se envían a una **función** y ésta devuelve el valor de la **hipotenusa**.



```
PROGRAMA Hipotenusa
  VARIABLES
    ENTERO catetoX, catetoY
    REAL hipotenusa
  INICIO
    ESCRIBIR "Introduce un cateto"
    LEER catetoX
    ESCRIBIR "Introduce el otro cateto"
    LEER catetoY
    hipotenusa = calcularHipotenusa(catetoX, catetoY)
    ESCRIBIR "La hipotenusa vale " + hipotenusa
  FIN
  FUNCION calcularHipotenusa(ENTERO x, ENTERO y) DEVUELVE REAL
  VARIABLES
    REAL resultado
  INICIO
    resultado = RAIZ((x*x)+(y*y))
    DEVOLVER resultado
  FIN
```

**Ejercicio 2** (2 puntos): Pasar el Ejercicio 1 a JAVA.

```

package es.studium.Hipotenusa;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Hipotenusa
{
    /* Cálculo de la hipotenusa de un triángulo */
    public static void main(String[] args) throws IOException
    {
        int catetoX, catetoY;
        double hipotenusa;
        BufferedReader lectura = new BufferedReader(new
InputStreamReader(System.in));
        System.out.print("Introduce un cateto:");
        catetoX = Double.parseDouble(lectura.readLine());
        System.out.print("Introduce el otro cateto: ");
        catetoY = Double.parseDouble(lectura.readLine());
        hipotenusa = calcularHipotenusa(catetoX, catetoY);
        System.out.println("La hipotenusa es: " + hipotenusa);
    }
    public static double calcularHipotenusa (int x, int y)
    {
        double resultado;
        resultado = Math.sqrt(Math.pow(x, 2)+Math.pow(y, 2));
        return (resultado);
    }
}

```

**Ejercicio 3** (2 puntos): Realizar un programa en PSEUDOCÓDIGO para crear la CLASE **VEHÍCULO** con los atributos **Descripción**, **Marca** y **Modelo**. Incluir todos los métodos necesarios.

**CLASE Vehiculo**

Atributos Privados:

CADENA[15] Marca

CADENA[15] Modelo

CADENA[50] Descripcion

Métodos(Declaración Pública):

//Constructores

Vehiculo()

Vehiculo(mar CADENA[15], mod CADENA[15], desc CADENA[50])

//Inspectores

pon\_marca(mar CADENA[15])

CADENA[15] dime\_marca()

pon\_modelo(CADENA[15])

```

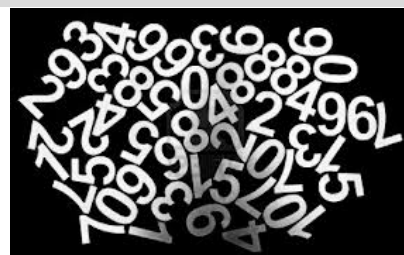
    CADENA[15] dime_modelo()
    pon_descripcion(CADENA[50])
    CADENA[50] dime_descripcion()
Métodos(Implementación privada):
    Vehiculo()
        INICIO
            Marca = ""
            Modelo = ""
            Descripcion = ""
        FIN
    Vehiculo(mar CADENA[15], mod CADENA[15], desc CADENA[50])
        INICIO
            Marca = mar
            Modelo = mod
            Descripcion = desc
        FIN
    pon_marca(mar CADENA[15])
        INICIO
            Marca = mar
        FIN
    CADENA[15] dime_marca()
        INICIO
            DEVOLVER Marca
        FIN
    pon_modelo(mod CADENA[15])
        INICIO
            Modelo = mod
        FIN
    CADENA[15] dime_modelo()
        INICIO
            DEVOLVER Modelo
        FIN
    pon_descripcion(desc CADENA[50])
        INICIO
            Descripcion = desc
        FIN
    CADENA[50] dime_descripcion()
        INICIO
            DEVOLVER Descripcion
        FIN
FIN CLASE

```

**Ejercicio 4** (1 puntos): Realizar un programa en PSEUDOCÓDIGO para crear la CLASE **MOTOCICLETA** con los atributos **Descripción**, **Marca**, **Modelo** y **Cilindrada**. Incluir todos los métodos necesarios.

```
CLASE Motocicleta HEREDA DE Vehiculo
  Atributos Privados:
    ENTERO Cilindrada
  Métodos(Declaración Pública):
    //Constructores
    Motocicleta()
    Motocicleta(mar CADENA[15], mod CADENA[15], desc CADENA[50],
cil ENTERO)
    //Inspectores
    pon_cilindrada(cil ENTERO)
    ENTERO dime_cilindrada()
  Métodos(Implementación privada):
    Motocicleta()
      INICIO
        Marca = ""
        Modelo = ""
        Descripcion = ""
        Cilindrada = 0
      FIN
    Motocicleta(mar CADENA[15], mod CADENA[15], desc CADENA[50],
cil ENTERO)
      INICIO
        Marca = mar
        Modelo = mod
        Descripcion = desc
        Cilindrada = cil
      FIN
    pon_cilindrada(cil ENTERO)
      INICIO
        Cilindrada = cil
      FIN
    ENTERO dime_cilindrada()
      INICIO
        DEVOLVER Cilindrada
      FIN
FIN CLASE
```

**Ejercicio 5** (3 puntos): Realizar un programa en JAVA que dada una tabla de 50 números aleatorios entre 1 y 100, nos pida un número entre 1 y 100 y nos muestre cuántos valores de la tabla son menores que el número dado, cuántos iguales y



cuántos mayores. Para simplificar el proceso, ANTES de nada ordenar la tabla por el método que menos os guste.

```
package es.studium.Tabla;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Random;
public class Tabla
{
    /* Tabla 100 números ordenar y calcular mayores, menores e iguales */
    public static void main(String[] args) throws IOException
    {
        BufferedReader lectura = new BufferedReader(new
InputStreamReader(System.in));
        int tabla[] = new int[100];
        int i, j, buscado, iguales, aux;
        Random aleatorio = new Random();
        // Rellenamos la tabla con los números aleatorios
        for(i=0; i<tabla.length; i++)
        {
            tabla[i] = (int)(aleatorio.nextDouble()*100+1);
        }
        // Ordenamos la tabla
        for(i=0; i<tabla.length-1; i++)
        {
            for(j=i+1; j<tabla.length; j++)
            {
                if(tabla[i]>tabla[j])
                {
                    aux = tabla[j];
                    tabla[j] = tabla[i];
                    tabla[i] = aux;
                }
            }
        }
        System.out.print("Dame el número a buscar: ");
        buscado = Integer.parseInt(lectura.readLine());
        i=0;
        while (tabla[i]<buscado)
        {
            i++;
        }
        System.out.println("Hay " + i + " números menores al buscado.");
        iguales = 0;
        while (tabla[i]== buscado)
        {
            i++;
        }
    }
}
```

```
        iguales ++;  
    }  
    System.out.println("Hay " + iguales + " números iguales al buscado.");  
    System.out.println("Hay " + (tabla.length-i) + " números mayores al buscado.");  
}  
}
```