



Tipos de triángulos

Problema 1

Escribe un programa completo (declaraciones, especificación y función main) que solicite al usuario las longitudes de los tres lados de un triángulo y que escriba en la pantalla si dicho triángulo es equilátero, isósceles o escaleno.

El programa debe presentar una forma de interacción con el usuario como la que se muestra en los ejemplos siguientes, que corresponden con ejecuciones **independientes** del programa:

```
Escriba las longitudes de los lados de un triángulo: 4.1 4.1 4.1  
El triángulo es equilátero.
```

```
Escriba las longitudes de los lados de un triángulo: 2 2 3  
El triángulo es isósceles.
```

```
Escriba las longitudes de los lados de un triángulo: 3.5 6 6  
El triángulo es isósceles.
```

```
Escriba las longitudes de los lados de un triángulo: 3 6 3  
El triángulo es isósceles.
```

```
Escriba las longitudes de los lados de un triángulo: 3 4 5  
El triángulo es escaleno.
```

```
Escriba las longitudes de los lados de un triángulo: 6 0 5  
Algún dato no es positivo.
```

```
Escriba las longitudes de los lados de un triángulo: -2 3 5  
Algún dato no es positivo.
```

(El texto resaltado en **negrita y subrayado** corresponde con posibles entradas suministradas por el usuario a través del teclado. El resto del texto representan escrituras del programa en la pantalla.)

Resuelve el problema de forma incremental, es decir, construyendo distintas versiones del programa que, poco a poco, vayan incorporando los requisitos solicitados en el enunciado. En particular, se sugiere seguir el siguiente proceso:

- Comienza escribiendo un programa que pida los tres datos al usuario y escriba en la pantalla si el triángulo de esos lados es equilátero o no lo es (es decir, escriba “El triángulo es equilátero” o “El triángulo no es equilátero”).
- Continúa modificando el programa para que, además de ser capaz de determinar si el triángulo es equilátero, determine también uno de los casos en los que es isósceles (por ejemplo, cuando los dos primeros lados son iguales).
- Continúa haciendo que el programa detecte los otros dos casos en los que el triángulo es isósceles (cuando los dos últimos lados son los iguales y cuando son iguales el primero y el tercero).
- Complétalo haciendo que escriba que es escaleno cuando no sea equilátero ni isósceles.
- Por último, añade el código necesario para que el programa compruebe si los tres datos suministrados por el usuario son positivos y haz que, en caso de que no lo sean, escriba el mensaje de error “Algún dato no es positivo”.



Años bisiestos

Problema 2

Escribe un programa completo (declaraciones, especificación y función main) que indique si un año, previamente solicitado al usuario y leído del teclado, es bisiesto o no. El programa debe presentar una forma de interacción con el usuario como la que se muestra en los dos ejemplos siguientes:

```
Escriba un año: 2016  
El año 2016 es bisiesto
```

```
Escriba un año: 2018  
El año 2018 no es bisiesto
```

Problema 3

Es posible que el programa que hayas planteado en el problema anterior sea incorrecto desde el punto de vista del dominio del problema que estamos resolviendo. Averigua cuándo un año es bisiesto o no.

Indica cuáles de los siguientes años han sido o serán bisiestos:

2000	2008	2016	2018	2019	2020
1900	2100				
1492	1500	1582	1600	1700	

(Habría sido conveniente que este ejercicio se hubiera hecho antes que el anterior. El **análisis** del problema que hay que resolver debe preceder al **diseño** e **implementación** de su solución)

Problema 4

Escribe una solución correcta al problema de saber si un año es bisiesto o no. En esta ocasión, apóyate en una función denominada `esBisiesto` que debes implementar y cuya especificación debe ser la siguiente:

```
/*  
 * Devuelve true si y solo si el año «agno» es bisiesto de acuerdo con el calendario gregoriano.  
 * El valor del parámetro «agno» debe ser posterior a 1582.  
 */  
bool esBisiesto(unsigned int agno) {  
    ...  
}
```

Modifica la función main del problema 1 para que haga uso de la función `esBisiesto` que has implementado.

Matrículas de honor

Problema 5

El artículo 14.2 del acuerdo de 22 de diciembre de 2010, del Consejo de Gobierno, por el que se aprueba el *Reglamento de normas de evaluación y aprendizaje de la Universidad de Zaragoza* establece:

La mención «Matrícula de Honor» podrá ser otorgada a estudiantes que hayan obtenido una calificación igual o superior a 9 en la asignatura. Su número no podrá exceder del 5% de los estudiantes matriculados en dicha asignatura en el correspondiente grupo docente, salvo cuando sea inferior a 20, en cuyo caso se podrá conceder una sola matrícula de honor.

Escribe una función C++ (especificación, cabecera y cuerpo) que determine el número de matrículas de honor que pueden otorgarse por grupo de docencia en la Universidad de Zaragoza en función del número de estudiantes matriculados en el mismo.