

1. Sean 4 variables enteras

$\text{int } m, j, p, v ;$   
que contienen respectivamente la edad de Miguel, Julio, Pablo y Vicente.

Expresar las siguientes afirmaciones utilizando operadores lógicos y relacionales

### Ejemplo

Miguel es mayor de edad.

Solución:  $m \geq 18$

- a) Miguel es menor de edad.
- b) Miguel es mayor que Julio
- c) Miguel es el más viejo.
- d) Miguel no es el más joven.
- e) Miguel y Julio son menores de edad
- f) Alguno de ellos es mayor de edad.
- g) Miguel y Julio son los más jóvenes.
- h) Entre todos tienen más de 100 años.
- i) Entre Miguel y Julio suman más edad que Pablo.
- j) Entre Miguel y Julio suman más edad que Pablo y Vicente juntos.
- k) Si los ordenamos por edades de menor a mayor, Julio es el segundo.
- l) Al menos uno de ellos es menor de edad.
- m) Todos son menores de edad.
- n) Solo dos de ellos son menores de edad.
- o) Al menos dos de ellos nacieron el mismo año.
- p) Al menos uno de ellos es menor que Julio

2. (Bisiesto) Escribir un método *public static boolean esBisiesto (int anyo)*, que devuelva si un año que se recibe como parámetro es bisiesto o no. Son bisiestos los años divisibles por 4, excepto los que son divisibles por 100, a no ser que sean divisibles por 400. Probar el método desde el método main.

3. Indicar qué mostrará por pantalla el siguiente fragmento de programa

```
int a = 2, b = 3, c;  
c = ++a * b++;  
System.out.format ("a = %d, b = %d, c = %d" , a, b, c);
```

4. (Switch) Implementar los métodos que se indican a continuación **utilizando la sentencia switch**. Usar el método main para probar los métodos.

- public static String nombreMes (int mes), que dado un número de mes, devuelve el nombre del mes correspondiente.
- public static String estacionDelAnyo (int mes), que dado un número de mes, devuelve el nombre de la estación correspondiente (“primavera”, “verano”, “otoño”, “invierno”)
- public static int numeroDiaSemana (String dia), que dado el nombre de un día de la semana (“lunes”, “martes”, ...), devuelve el número de día de la semana correspondiente (1,2,...)
- public static String nombreDiaSemana (int dia), que dado el número de día de la semana, devuelve “lunes”, “martes”, etc.
- public static int diasDelMes (int mes), que dado un número de mes devuelve cuántos días tiene el mes. Se supone que febrero tiene 28
- public static int diasDelMes (int mes, int anyo), que dado un número de mes y de año devuelve cuántos días tiene el mes. Febrero tendrá 28 o 29 días dependiendo de si el año es bisiesto o no.

5. (LeerEntero) Escribir el siguiente método y probarlo desde el main

- `public static int leeEnteroPositivo ( )`, que lea de teclado un número entero. Si el usuario introduce un número menor o igual a cero, el método mostrará un mensaje de error y se lo volverá a pedir (bucle).
6. (Containers) La capacidad de un buque que transporta containers está limitada tanto por la cantidad de containers como por el peso, pudiendo transportar un máximo de 100 containers y un máximo de 700 toneladas. Hacer un programa en el que se vaya introduciendo el peso de los containers (en toneladas) a medida que se cargan en el barco, hasta que se llegue al máximo de capacidad. Mostrar al final la cantidad de containers cargados y el peso total. En el momento en que se desee cargar un container que haga que la carga total supere las 700 toneladas, se dará por finalizada la carga, aunque pudieran existir containers menos pesados con posibilidad de ser cargados.
  7. (Notas) Realizar un programa que permita introducir las notas de un examen de los alumnos de un curso. El usuario irá introduciendo las notas una tras otra. Se considerará finalizado el proceso de introducción de notas cuando el usuario introduzca una nota negativa. Al final, el programa le dirá:
    - El número de notas introducidas.
    - El número de aprobados (mayor o igual a 5 puntos)
    - La nota media
  8. (CaraOCruz) Escribir un programa que simule que se lanza una moneda al aire una y otra vez. En cada iteración del bucle habrá que mostrar cuántas veces ha salido cara, cuántas veces ha salido cruz, porcentaje de caras y porcentaje de cruces. El proceso se detendrá cuando la diferencia entre el porcentaje de caras y de cruces sea menor a 0.001.
  9. (BlackJack) Escribir un programa que simule una jugada de Blackjack (simplificado). El jugador tiene que pedir cartas para intentar sumar o acercarse lo más posible a 20 puntos, sin pasarse. Puede dejar de pedir cartas cuando lo desee.
    - El jugador parte con una puntuación de 0.
    - De forma repetitiva:
      - Se le pregunta si quiere otra carta.

- Si dice que si, el programa le da otra carta (número aleatorio entre 1 y 10), que se añade a su puntuación.
- El programa termina cuando el jugador indica que no quiere otra carta o cuando su puntuación iguala o supera el valor 20.

**10.(AdivinarNumero)** Escribir un programa en el que el usuario trate de adivinar un número entre 1 y 100 que previamente ha generado aleatoriamente el programa:

- El programa genera un número aleatorio entre 1 y 100. Lo mostraremos por pantalla para poder comprobar si el programa funciona bien o no
- De forma repetitiva:
  - El usuario intenta adivinar el número.
  - El programa le indica si ha acertado, si se ha quedado corto o si se ha pasado.
- El programa finaliza cuando el usuario adivina el número

**Ejemplo de ejecución:**

```
Numero a adivinar: 80

Adivina el número: 20
Te has quedado corto
Adivina el número: 90
Te has pasado
Adivina el numero: 60
Te has quedado corto
Adivina el numero: 80
Has acertado
```

**11.(AdivinarNumero2)** Repetir el programa anterior dando al usuario un máximo de intentos para adivinar el número. El programa finalizará si el usuario adivina el número o si no lo ha adivinado en 5 intentos.

**12.El funcionamiento de un almacén es el siguiente:**

- Su capacidad máxima es de 500 unidades.
- El stock mínimo es de 100 unidades. Cuando se baja de dicho stock mínimo se comprará mercancía para llenar el almacén de nuevo.

Hacer un programa que simule el funcionamiento del almacén, suponiendo que inicialmente está al máximo de su capacidad (500). El programa solicitará al usuario la cantidad que se quiere servir a los clientes cada día y responderá al usuario de alguna de las siguientes formas:

- Si el stock es insuficiente para servir el pedido, se mostrará: “Stock insuficiente”
- Si el stock es suficiente para servir el pedido, se descontará la cantidad servida al stock y se mostrará éste por pantalla. Además, si se rebasa el stock mínimo, se comprará mercancía para llenar el almacén de nuevo y se mostrará la cantidad comprada.

El programa finalizará cuando el usuario introduzca un cero como cantidad a servir.

Ejemplo de ejecución:

Introduzca cantidad a servir el día 1: 200

Pedido servido.

Queda en almacén: 300

Introduzca cantidad a servir el día 2: 120

Pedido servido.

Queda en almacén: 180

Introduzca cantidad a servir el día 3: 200

Stock insuficiente.

Queda en almacén: 180

Introduzca cantidad a servir el día 4: 120

Pedido servido.

Rebasado stock mínimo. Se compra 440 unidades.

Queda en almacén: 500

Introduzca cantidad a servir el día 5: 100

Pedido servido.

Queda en almacén: 400

Introduzca cantidad a servir el día 6: 0

Fin del programa.