1. Indica qué **identificadores** son válidos. Si no son válidos indica el error.

```
a) X0AB       b) Else     c) e2

d) e_2        e) 2e2      f) OK
```

2. Indica si los siguientes **literales** son válidos o no. Si no son válidos indica el motivo. Si son válidos indica su tipo. Si el tipo es numérico indica en qué sistema de numeración están escritos.

```
a) X0AB     b) 0612L    c) '\''       d) e2         e) 2e2

f) 2e2f     g) 2_e2     h) '\c0077'   i) '"'        j) 1.23FE2

k) +2.2e+3  l) '\\"'    m) "Nombre:\" n) 0xF        o) "'A'''B'"

p) "//\"/'" q) "|/g|"   r) 0x2.5      s) "else"     t) 7e-2

u) 01
```

3. Explica qué se muestra por pantalla

```
a) int x = 14, y = 3;
   S.o.p(y <= x / 4 && y > x % 3 ? x++ + y:--x + y);

b) int x = 7, y = 3;
   S.o.p(x-y / 2 % 3 >= y ? x + 2 * y : x / y + 2);

c) int a = 1, b = 2, x = 5, y = 2;
   a = 2.5 <= x / y ? ++b + b : a + ++b;
   S.o.p((100 + a % 10 == 4) ? "verdadero" : "falso");

d) char x = 'p', y = 'c', z;
   int b = 3, c = b++;
   z = c == b ? x++ : y++;
   S.o.p(x + " " + y + " " + z);

e) boolean X = true, Y = true, Z = false;
   double i = !(X || Y && Z) ? 3 / 2 : 5 / 2;
   S.o.p(i);

f) boolean X = true, Y = false;
   char a = 'x', b = a--, c = a != b || !(X && Y) ? a : b;
   S.o.p(a + " " + b + " " + c);

g) int a = 6, b = 5, c = -2;
   int d = a % 2 == 0 && b % 2 == 0 ? a+c : a+c > b ? a-c : 0;
   S.o.p(d);

h) int a = 7;
   double b = 2;
   double c = a / b == 3 ? ++a : a % 4 != 0 ? a + b : ++b;
   S.o.p(c % 3 == 0 ? c % 6 == 0 ? "A" : "B" : "C");
```

SOLUCIONES

1. Indica qué identificadores son válidos. Si no son válidos indica el error.

a) X0AB Válido

b) Else Válido

c) e2 Válido

d) e_2 Válido

e) 2e2 No válido. Empieza por dígito

f) OK Válido

2. Indica si los siguientes **literales** son válidos o no. Si no son válidos indica el motivo. Si son válidos indica su tipo. Si el tipo es numérico indica en qué sistema de numeración están escritos.

a) X0AB NO VÁLIDO.

b) 0612L VÁLIDO long OCTAL

c) '\'' VÁLIDO \' ES UNA SECUENCIA DE ESCAPE

d) e2 NO VÁLIDO. FALTA LA BASE

e) 2e2 VÁLIDO. double decimal

f) 2e2f VÁLIDO float decimal

g) 2_e2 NO VÁLIDO. _ NO PUEDE SEPARAR UN DÍGITO DE LA e

h) '\c0077' NO VÁLIDO. \c no es una secuencia de escape
debería ser \u

i) '"' Válido. char

j) 1.23FE2 NO VÁLIDO. F debería ir al final

k) +2.2e+3 VÁLIDO. double decimal

l) '\\"' NO VÁLIDO. DEBERIA SER '\\' Ó '\"'

m) "Nombre:\" NO VÁLIDO. DEBERÍA SER "Nombre:\\" Ó "Nombre:\""

n) 0xF VÁLIDO int hexadecimal

o) "'A'''B'" VÁLIDO. STRING

p) "//\"/'" VÁLIDO. STRING

q) "|/g|" VÁLIDO. STRING

r) 0x2.5 NO VÁLIDO. UN ENTERO NO PUEDE LLEVAR .

s) "else" VÁLIDO. STRING

t) 7e-2 VÁLIDO. double. decimal

u) 01 VÁLIDO. int. Octal

3. Explica qué se muestra por pantalla

```
a) int x = 14, y = 3;
   S.o.p(y <=  x / 4 && y > x  % 3 ? x++ + y:--x + y);
         3 <= 14 / 4 && 3 > 14 % 3
         3 <=    3   && 3 >    2
           T         &&   T
                     T
   La condición es cierta ->
   se muestra el resultado de x++ + y
   Primero se suma  x + y después se incrementa x
                  14 + 3
   Muestra ->  17
   El valor de x se incrementa después de realizar la suma
```

```
b) int x = 7, y = 3;
   S.o.p(x - y / 2 % 3 >= y ? x + 2 * y : x / y + 2);
         7 - 3 / 2 % 3 >= 3
         7 -   1   % 3 >= 3
         7 -       1   >= 3
                   6   >= 3
                       T
    La condición es cierta
    se muestra el resultado de x + 2 * y
                               7 + 2 * 3
                               7 +   6
    Muestra -> 13
```

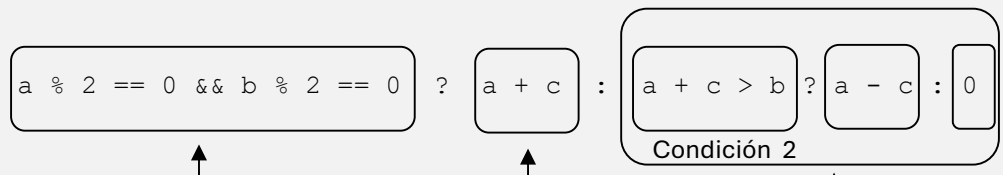
```
c) int a = 1, b = 2, x = 5, y = 2;
   a = 2.5 <= x / y ? ++b + b : a + ++b;
       2.5 <= 5 / 2
       2.5 <=   2
           F
       La condición es falsa -> a = a + ++b;
       Primero se incrementa el valor de b(3)
       después se suma con a(1) -> a = 4
   S.o.p((100 + a % 10 == 4) ? "verdadero" : "falso");
          100 + 4 % 10 == 4
          100 +   4    == 4
             104       == 4
                        F
          La condición es falsa -> Muestra: falso
```

```
d) char x = 'p', y = 'c', z;
   int b = 3, c = b++;
              c = 3  b = 4 -> primero c = b y después b++
   z = c == b ? x++ : y++;
       3 == 4
         F
       La condición es falsa -> z = y++
       primero se realiza z = y -> z = 'c'
       después se incrementa y ->  y = 'd'
   S.o.p(x + " " + y + " " + z);
   Muestra p d c
```

```
e) boolean X = true, Y = true, Z = false;
   double i = !(X || Y && Z) ? 3 / 2 : 5 / 2;
              !(T || T && F)
              !(T ||    F  )
              !(   T       )
              F
        La condición es falsa -> i = 5 / 2 -> i = 2.0
   S.o.p(i);
   Muestra 2.0

f) boolean X = true, Y = false;
   char a = 'x', b = a--,
                 primero se realiza b = a -> b = 'x'
                 después se decrementa a -> a = 'w'
   c = a  !=  b  || !(X && Y) ? a : b;
      'w' != 'x' || !(T && F)
           T     || !(T && F)
           T     || !(  F   )
           T     ||   T
                  T
   La condición es cierta -> c = a -> c = 'w'
   S.o.p(a + " " + b + " " + c);
   Muestra w x w

g) int a = 6, b = 5, c = -2;
   int d = a % 2 == 0 && b % 2 == 0 ? a + c : a + c > b ? a-c : 0;
              instrucción con operadores condicionales anidados
```



Condición principal

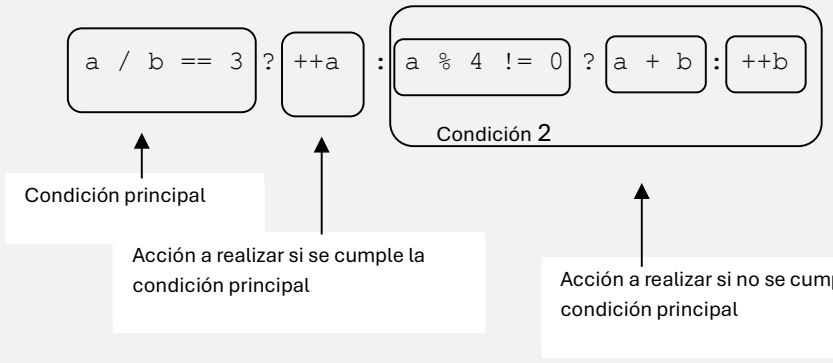
Acción a realizar si se cumple la condición principal

Acción a realizar si no se cumple la condición principal

```
          a % 2 == 0 && b % 2 == 0
          6 % 2 == 0 && 5 % 2 == 0
            0   == 0 &&   1   == 0
                T    &&       F
                     F
    La condición principal es falsa
    int d =  a + c  > b ? a - c : 0
             6 + -2 > 5
               4    > 5
                    F
    La condición 2 es falsa por lo tanto -> d = 0
   System.out.println(d);
           Muestra -> 0
```

h)
```
int a = 7;
double b = 2;
double c = a / b == 3 ? ++a : a % 4 != 0 ? a + b : ++b;
```

instrucción con operadores condicionales anidados




```
a / b   == 3
7 / 2.0 == 3
  3.5   == 3
        F
```

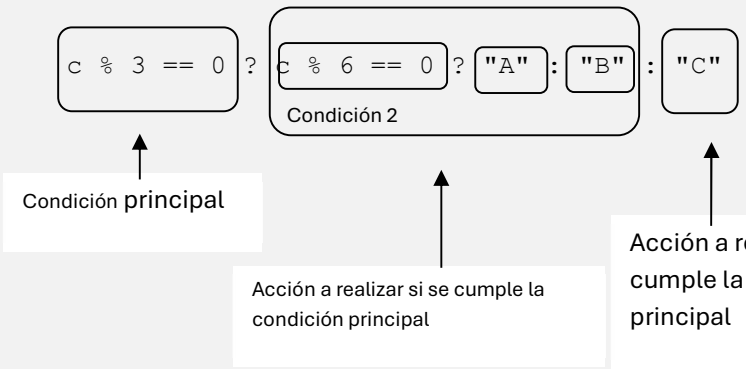
La condición principal es false -> double c = a % 4 != 0 ? a + b : ++b

```
7 % 4 != 0
  3   != 0
      T
```

La condición 2 es cierta -> double c = a + b -> c = 9.0

```
System.out.println(c % 3 == 0 ? c % 6 == 0 ? "A" : "B" : "C");
```

otra instrucción con operadores condicionales anidados




```
  c % 3 == 0
9.0 % 3 == 0
  0.0   == 0
        T
```

La condición principal es cierta.

Se muestra por pantalla el resultado de evaluar c % 6 == 0 ? "A" : "B"

```
9.0 % 6 == 0
  3.0   == 0
        F
```

La condición 2 es falsa -> Muestra: B