EJERCICIO DE OPERADORES LOGICOS

Ejemplos:

En el siguiente ejemplo hay dos expresiones relacionales que devolverán un valor lógico: a > b y b < 0. Como están entre paréntesis se evaluarán primero, y sus valores resultantes serán lo que evalúe el operador &&:

```
int a = 5;

int b = -1;

(a &gt; b) &amp;&amp; (b < 0); //true
```

a > b: El resultado es true puesto que la variable a tiene valor 5 y la variable b tiene valor -1. Se cumple 5 > -1.

b < 0: El resultado es true puesto que la variable b tiene valor -1. Se cumple -1 < 0.

Como a>b es true y b<0 es true, la expresión (a > b) && (b < 0) se evaluará como true && true, lo que dará como resultado true.

En este segundo ejemplo uno de los operandos tiene valor false:

```
int a = 5;

int b = -1;

(a == 0) &amp;amp;&amp;amp; (b == -1); //false
```

a == 0: El resultado es false puesto que la variable a tiene valor 5. 5 == 0 no es cierto.

b == -1: El resultado es true ya que b tiene valor -1. -1 == -1 es cierto.

Como a==0 es false y b==-1 es true, la expresión (a == 0) && (b == -1) se evaluará como false && true, lo que dará como resultado false porque no son true los dos operandos.

En este caso, como el primer operando es false, se produce cortocircuito y no se evaluaría el segundo, retornando como valor inmediato false.

Ejemplos:

En el siguiente ejemplo hay dos expresiones relacionales que devolverán un valor lógico: b < 0 y a < b. Como están entre paréntesis se evaluarán primero, y sus valores resultantes serán los que evalúe el operador ||:

```
int a = 5;

int b = -1;

(b &gt; 0) || (a < b); //true
```

b < 0: El resultado es true puesto que la variable b tiene valor -1. Se cumple -1 < 0.

a < b: El resultado es false puesto que la variable a tiene valor 5 y la variable b tiene valor -1. No se cumple 5 < -1.

Ya que b < 0 es true y a < b es false, la expresión (b < 0) || (a < b) se evaluará como true || false, lo que dará como resultado true, porque al menos uno de los operandos es true.

En este caso, como el primer operando es true, se produce cortocircuito y no se evaluaría el segundo, retornando como valor inmediato true.

En este segundo ejemplo los dos operandos tienen valor false:

```
int a = 5;

int b = -1;

(a == 0) &amp;amp;&amp;amp; (b == a); //false
```

a == 0: El resultado es false puesto que la variable a tiene valor 5. 5 == 0 no es cierto.

b == a: El resultado es false ya que a tiene valor 5 y b tiene valor -1. 5 == -1 es falso.

Como a == 0 es false y b == a es false, la expresión (a == 0) && (b == a) se evaluará como false || false, lo que dará como resultado false porque no hay al menos un operando con valor true.

Ejemplos:

En este primer ejemplo el operando sobre el que se aplica el operador es una expresión relacional que devuelve true:

```
int a = 9;

 !(a == 9); //false
```

a == 9: La variable a toma valor 9. El resultado de la evaluación es true porque se cumple 9 == 9.

Como a == 9 es true la evaluación de !(a == 9) será !true, lo que da como resultado false.

En este otro ejemplo el operando tiene valor false:

```
int a = -1;

int b = 0;
```

```
!(a &gt; b); //true
```

a > b: La variable a toma valor -1 y la variable b toma valor 0. El resultado de la evaluación es false porque no se cumple -1 > 0.

Como a > b es false la evaluación de !(a > b) será !false, lo que da como resultado la inversión del valor, que es true.