

Background:

Computer scientists use the laws of boolean logic to understand and simplify the design of logic circuits. Boolean logic can also be used to simplify software. Often, complicated expressions can be reduced to much simpler calculations. Some of the important laws of logic are:

The identity law:

$$\begin{aligned} p \ \&\& \ \text{true} &= p \\ p \ \|\ \text{false} &= p \\ p \ \|\ \text{true} &= \text{true} \\ p \ \&\& \ \text{false} &= \text{false} \end{aligned}$$

The inverse law:

$$\begin{aligned} p \ \&\& \ !p &= \text{false} \\ p \ \|\ !p &= \text{true} \end{aligned}$$

DeMorgan's law:

$$\begin{aligned} !(p \ \&\& \ q) &= (!p) \ \|\ (!q) \\ !(p \ \|\ q) &= (!p) \ \&\& \ (!q) \end{aligned}$$

The absorption rule:

$$\begin{aligned} p \ \&\& \ (p \ \|\ q) &= p \\ p \ \|\ (p \ \&\& \ q) &= p \end{aligned}$$

The idempotent law:

$$\begin{aligned} (p \ \&\& \ p) &= p \\ (p \ \|\ p) &= p \end{aligned}$$

The distributive law:

$$\begin{aligned} p \ \&\& \ (q \ \|\ r) &= (p \ \&\& \ q) \ \|\ (p \ \&\& \ r) \\ p \ \|\ (q \ \&\& \ r) &= (p \ \|\ q) \ \&\& \ (p \ \|\ r) \end{aligned}$$

Problem:

Using the laws of logic (and a little arithmetic/high school algebra), simplify the following C++ function as much as possible.

```
int computeValue(int x, int y){
    int z;
    if (((x==y && y==x) && (x==y || x==y)) || (x==y && x==x)) &&
        (x != y) ){
        z = (x+y)*(x+y)+30;
    }
    else{
        z = (10*(x+y) + 30)/10 ;
        z -= (x+y);
    }
    return z;
}
```

Solutions:

To submit a solution to the problem of the month, e-mail your answer and a complete, rigorous justification to robert@narnia.homeunix.com. I will give a cash prize of \$1 to the student who submits the first correct and complete solution. Subsequent solvers will be recognized on the web site, but not awarded a cash prize. This problem is also available online at <http://narnia.homeunix.com/~robert/PoTM/>