CvP Practice Class 5

Chapter 7: Problem Set

7. Describe a situation in which the add operator in a programming language would not be commutative.

8. Describe a situation in which the add operator in a programming language would not be associative.

9. Assume the following rules of associativity and precedence for expressions:

Precedence :	*Highest*	*, /, **not**
		+, -, &, **mod**
		- (unary)
		=, /=, <, ≤, ≥, >
		and
	Lowest	**or**, **xor**
Associativity	*Left to right*	

Show the order of evaluation of the following expressions by parenthesizing all subexpressions and placing a superscript on the right parenthesis to indicate order. For example, for the expression

$$a + b * c + d$$

the order of evaluation would be represented as

$$(a + (b * c)^1)^2 + d)^3$$

(a) $a * b - 1 + c$

(b) $a * (b - 1)/c \bmod d$

(c) $(a - b)/c \ \&\ (d * e/a - 3)$

(d) $-a$ or $c = d$ and e

(e) $a > b$ xor c or $d \leq 17$

(f) $-a + b$

10. Show the order of evaluation of the expressions of Problem 9, assuming that there are no precedence rules and all operators associate right to left.

11. Write a BNF description of the precedence and associativity rules defined for the expressions in Problem 9. Assume the only operands are the names *a, b, c, d* and *e*.

13. Let the function *fun* be defined as

```
int fun(int *k) {
  *k += 4;
  return 3 * (*k) - 1;
}
```

Suppose *fun* is used in a program as follows:

```
void main() {
  int i = 10, j=10, sum1, sum2;
  sum1 = (i / 2) + fun(&i);
  sum2 = fun(&j) + (j / 2);
}
```

What are the values of *sum1* and *sum2*

(a) if the operands in the expressions are evaluated left to right?

(b) if the operands in the expressions are evaluated right to left?