

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

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

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

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

- $a * b - 1 + c$
- $a * (b - 1) / c \text{ mod } d$
- $(a - b) / c \& (d * e / a - 3)$
- $\neg a \text{ or } c = d \text{ and } e$
- $a > b \text{ xor } c \text{ or } d \leq 17$
- $\neg a + b$

9.

- $((a * b)^1 - 1)^2 + c)^3$
- $((a * (b - 1)^1)^2 / c)^3 \text{ mod } d)^4$
- $((a - b)^1 / c)^5 \& (((d * e)^2 / a)^3 - 3)^4)^6$
- $((\neg a)^1 \text{ or } ((c = d)^2 \text{ and } e)^3)^4$
- $((a > b)^1 \text{ xor } c)^3 \text{ or } (d \leq 17)^2)^4$
- $\neg(a + b)^1)^2$

Joseph Camacho-Terrazas

Chapter 7 Problem Set

10/16/2020

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. operands in the expressions are evaluated left to right?
- b. operands in the expressions are evaluated right to left?

13.

- a. **sum1 = 46, sum2 = 48**
- b. **sum1 = 48, sum2 = 46**

19. Consider the following C program:

```
int fun(int *i) {
    *i += 5;
    return 4;
}
```

Expressions and Assignment Statements

```
void main() {
    int x = 3;
    x = x + fun(&x);
}
```

What is the value of `x` after the assignment statement in `main`, assuming

- a. operands are evaluated left to right.
- b. operands are evaluated right to left.

19.

- a. **x = 7**
- b. **x = 12**