1- 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 mod d`

c. `(a - b) / c & (d * e / a - 3)`

d. `-a or c = d and e`

e. `a > b xor c or d <= 17`

f. `-a + b`

2-

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?

3-

Consider the following C program:

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