- **Homework**

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   ⇨   ((a + (b * c)1)2 + d)3
```

① `a * b - 1 + c`
② `a * (b - 1) / c mod d`
③ `(a - b) / c & (d * e / a - 3)`
④ `-a or c = d and e`
⑤ `a > b xor c or d <= 17`
⑥ `-a + b`

2. Show the order of evaluation of the expressions of Problem 1, assuming that there are no precedence rules and all operators associate right to left.

3. Let the function **fun** and its usage be defined as

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

```
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

① if the operands in the expressions are evaluated left to right ?

② if the operands in the expressions are evaluated right to left ?

4. 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

① operands are evaluated left to right.

② operands are evaluated right to left.

5. Let the function **fun** and its usage be defined as

```
int a, b;
main() {
        a = 10;
        b = a + fun();
        printf("With the function call on the right, ");
        printf(" b is: %d\n", b);
        a = 10;
        b = fun() + a;
        printf("With the function call on the left, ");
        printf(" b is: %d\n", b);
}

fun() {
        a = a + 10;
        return(a);
}
```

Explain the results.