This document is written Xiao Liang with the help from Evan and Joanne. If you have found any mistake, please let me know.

This tutorial should go through Problem Sets 20, 21, 22. Possibly the solution of as05.

20.1

a)

```
#define MIN(a,b)  a < b ? a : b

long i = MIN(10, 20);
long j = MIN(10, 20) + 1;
```

Both `i` and `j` are 10 in this case. This is because when you expand the expression of `j`, you will get `long j = 10 < 20 ? 10 : 20 + 1` which is same `long j = 10 < 20 ? 10 : 21`. This precedence thing is the main part of this question.

b)

```
#define MIN(a,b)  a < b ? a : b

long i = 10;
long j = 20;
long k = MIN(j, i++);
```

When `long k = MIN(j, i++)` gets expanded, you will get `long k = j < i++ ? j : i++`. In this case, the evaluation will be evaluated in the following order:

```
long k = j < i++ ? j : i++
-> j < i++ // after execution: i = 11
-> i++ // after execution: i = 12
-> k = 11
```

At the end, `i` will be 12 and `k` will be 11 .

20.2

When putting arguments into macro function, think of it as simply string substitution. The problem is when `SWAP` in called in a scope contains variable `temp` , this will have duplicated declaration syntax error.

21.1

Yes. The possible values of `x % 2` are 0, 1, and -1 (According to C11 standard, `(a / b) * b + a % b == a` so `x % 2` **can be negative**). When `x % 2 == -1` , the assertion fails.

21.2

Ask them to do so if you wish. Would be interesting to see what assertion they will add.

22.1

$$O(\ln n) == O(\log_{10} n) << O(\sqrt{n}) << O(n) << O(n \ln n)$$
$$<< O(n^2) << O(n^4) << O(2^n) << O(e^n) << O(n!)$$

22.2

a)

$$O(n * n) = O(n^2)$$

b)

$$O(\log_2 n * \log_2 n) = O((\log_2 n)^2)$$

c)

$$O(2 + 2^2 + 2^3 ... + 2^n)$$
$$= O(\frac{2 * (1 - 2^n)}{1 - 2})$$
$$= O(\frac{2 - 2^{n+1}}{-1})$$
$$= O(2^{n+1} - 2))$$
$$= O(2^{n+1})$$
$$= O(2 * 2^n)$$
$$= O(2^n)$$