RECURSION

Lab Instructor : Jean Lai
WHAT IS RECURSION?

- Process of repeating an action in a similar way in a non-iterative manner.
- Purpose is to divide the problem into smaller problems.
- By solving one or more smaller problems, the solutions of the smaller problems will be used to solve the original problem.
RECURSIVE FUNCTION

- Function calls itself repeatedly.
- Each repeated call gets a smaller version of original problem.
- Stops when it reaches the base case.

- Classic examples of recursion
  - i.e. Fibonacci sequence, factorial, GCD
**GENERAL FORM**

Base case (stopping case)
- One or more cases that includes no recursive call.

Smaller Problem
- One or more cases that includes one or more recursive calls to itself.
- In each recursive call, the input size decreases.
Factorial Recursion

```c
int factorial (int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n-1);
}
```

Factorial(4)
- 4 * Factorial(3)
  - 3 * Factorial(2)
    - 2 * Factorial(1)
      - 1

*Base case*
No recursion calls!

*Smaller problem*
with smaller input size.
int Fibonacci (int n) {

    if (n <= 1)
        return n;
    else
        return Fibonacci(n-1) + Fibonacci(n-2);
}

Base case
No recursion calls!

Smaller problem
with smaller input size.