**Chapter 02 재귀(Recursion)**

* 1. **함수의 재귀적 호출의 이해**

〮 재귀함수 -> 함수 호출되면 해당 함수의 복사본을 만들어서 실행하는 구조

-> 정의 시 탈출조건 구성하는 것 매우 중요

**02-2 재귀의 활용**

〮 어떤 프로그램을 재귀적 사고를 가지고 code로 작성 -> 여기서 끝날 수 있어야함.

Code를 보고 호출관계와 호출 순서를 굳이 구하려 드는 것 X

Ex) 피보나치 수열

- 이진탐색 알고리즘 -> 수식적으로 표현하기는 부적절, but 논리자체는 재귀적이므로 재귀함수 기반으로 구현 가능

순서: 패턴(내용) 보고 재귀적임 판단 -> 코드 작성

**02-3 하노이 타워**

〮 목표: 재귀적인 사고를 통해 하노이 타워 이해

〮 문제 -> 해결책 -> code! (여기서 어렵다고 코드 파헤치는 순간 더 여려워진다.)

〮 막대에 A에 꽂혀있는 원반 n개를 막대 C로 옮기는 과정

( HanoiTowerMove (num-1, from, by, to); ) <- 기본골격

1. 작은 원반 n-1개(맨 아래의 원반을 제외한 나머지 원반을) A에서 b로 이동

( HanoiTowerMove (num, from, to, by); )

1. 큰 원반(맨 아래의 원반) 1개를 A에서 C로 이동

( printf(….); )

1. 작은 원반(위의 1단꼐에서 옮기 원반) n-1개를 B에서 C로 이동

( HanoiTowerMove (num-1, by , from , to); )

\*탈출 조건도 세우기