Lab1: Recursion 연습

1) **Recursive & Iterative Addition**

- ***For Recursive Addition*: (순환알고리즘 덧셈)**

"*n+m* ➔ *(n-1)+(m+1)*" At each step, subtract 1 from n and add 1 to m until "i is 0". Then return m.

Input: integers m ≥ 0, n ≥ 0 Output: m + n

Function ***sum (n, m)***
If n = 0 then return (m)
else return ***sum (***pred(n), succ(m)) // pred -> -1, succ->+1

예) **입력: 임의의 두 숫자 3, 7**

출력: add(3, 7)→ add(2, 8)→ add(1, 9)→ add(0, 10)

sum= 10

- ***For Iterative Addition: (반복문 사용)***

Function sum (n, m)
While n > 0 do
{ n := pred(n); m := succ(m); }
Return (m)

예) **입력: 임의의 두 숫자 2, 3**

출력: n=2 m=3; n=1 m=4; n=0 m=5;

sum= 5

2) Sum of numbers (Recursive algorithm 사용할 것)

알고리즘: Factorial 의 순환 알고리즘을 덧셈으로 변환.

예) 입력: 임의의 숫자 3

출력: 3+2+1 = 6

Extra Credit) Find FIBONACCI number

Recursive Algorithm:

```
Fibo( n ) {
If n=0, return 0
Else if n = 1, return 1
Else  return (fibo (n-1) + fibo(n-2))
}
```

Condition:

1) input n from keyboard
2) print out only the final value