

Homework 1  
SNU 4190.310, Fall 2017  
Kwangkeun Yi  
**Due: 9/14, 24:00**

**Exercise 1** “리스트합”

큰 순서대로(descending order) 나열된 정수 리스트 두개를 받아서 하나의 순서 리스트로 만드는 함수

`merge: int list * int list -> int list`

를 정의하세요. 리스트에는 같은 정수가 반복해서 들어있지 않습니다. □

**Exercise 2** “씨그마”

우리가 중고등 수학시간에 술하게 썼던 다음의 “씨그마”를 OCaml로 정의하세요:

$$\sum_{n=a}^b f(n)$$

씨그마의 타입은

`sigma : int * int * (int -> int) -> int.`

즉, `sigma(a,b,f)`로 표현하면  $\sum_{n=a}^b f(n)$ 과 같도록. □

**Exercise 3** “반복기”

다음함수 `iter`를 정의하세요:

$$\text{iter}(n, f) = \underbrace{f \circ \cdots \circ f}_n$$

이 때,  $n = 0$ 이면 아무일을 하지 않는(identity) 함수를 내 놓고, 양수이면 그 만큼  $f$ 를 반복해서 적용하는 함수를 내 놓습니다. 그래서,

`iter(n, function x -> 2+x) 0`

은  $2 \times n$ 을 내 놓게됩니다. □

**Exercise 4 “자연수”**

자연수 `nat` 는 다음과 같이 정의될 수 있다:

```
type nat = ZERO | SUCC of nat
```

두 자연수를 받아서 그 합/곱에 해당하는 자연수를 만드는 두 함수

```
natadd : nat * nat -> nat  
natmul : nat * nat -> nat
```

를 정의하세요. □