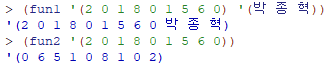
**1. 다음 scheme함수는 각기 무엇을 하는지 설명하고, DrRacket에서 실제 데이터를 가지고 실행한 결과를 캡쳐하여 보여라.**

이때 테스트 데이터는 본인의 영문이름이나 학번을 사용한다.

**가)**

fun1 : 리스트 2개를 매개변수로 받아서 두번째 리스트를 첫번째 리스트의 뒤에 합치는 함수

fun2 : 리스트 1개를 매개변수로 받아서 리스트 원소의 순서를 반대로 만드는 함수



**나)**

fun : 리스트 1개를 매개변수로 받아서 각 원소들을 제곱시키는 함수



**다)**

y : 리스트 lis에서 원소 s부터 맨 끝 원소까지 출력



**라)**

x : 리스트 1개를 매개변수로 받아 리스트 원소의 개수를 반환하는 함수



**2. 다음 scheme함수를 tail-recursive function으로 재작성하라.**

**tail-recursive**

(define (doit x acc)

(if (zero? x)

acc

(doit (sub1 x) (+ x acc))))

결과 :



**3. 아래 코드를 고려해 보자.**

(let ((a 1))

(let ((f (lambda ()

(display a))))

(let ((a 2))

(f))))

가) static scoping rules이라면 출력되는 값은? ( 1 )

나) dynamic scoping rules이라면 출력되는 값은? ( 2 )

**4. 다음 설명대로 수행하는 함수를 완성하여 정의하고 테스트하라. 본인의 학번이나 영문이름을 데이터로 적용하여 결과를 캡쳐하여 보여라.**

가) 리스트에서 마지막 원소를 찾는 함수

(define (final List)

(cond ((null? (cdr List)) (car List))

(else (final (cdr List)))))



나) 리스트에서 처음 n개의 원소를 제거해 주는 함수

(define (del-list List n)

(list-tail List n))

