

Homework 5
SNU 4190.310, 2010 가을
Kwangkeun Yi
Due: 10/30(Sat) 24:00

Exercise 1 (30pts) “Lambda Calculus”

람다 계산법에서 normal-order reduction 룰(강의 슬라이드)을 따르는 계산기

```
reduce: lexp -> lexp
```

를 구현하라. 타입 lexp는 아래와 같고, TA팀이 lexp 파서를 제공할 것이다.

```
type lexp = Id of string  
          | Lam of string * lexp  
          | App of lexp * lexp
```

□

Exercise 2 (30pts) “Encoding”

아래의 적극적인(eager-evaluation, or call-by-value) 프로그래밍 언어 M_0 를

생각하자:

e	::=	n	natural number
		id	identifier
		$fn\ id\ =>\ e$	function
		ee	application
		$rec\ id\ id\ =>\ e$	recursive function
		$ifzero\ e\ then\ e\ else\ e$	branch
		$e\ op\ e$	infix binary operation
		(e, e)	pair
		$e.1$	first component
		$e.2$	second component
op	::=	$+ - and$	
id			alpha-numeric identifier

위의 언어로 짜여진 제대로 돌아가는 프로그램을 람다식으로 번역하는 함수:

encode: mexp -> lexp

를 정의하라. 조건은: 번역된 결과를 Exercise 1에서 만든 reduce로 실행시켜서 원래 M_0 프로그램의 결과와 같은 결과를 내놔야 한다. 이 때, 번역할 프로그램은 항상 자연수를 최종적으로 계산하는 프로그램으로 제한한다. TA팀이 mexp 파서를 제공할 것이다.

```
type mexp = Num of int
          | Var of string
          | Fn of string * exp
          | Rec of string * string * exp
          | Ifz of exp * exp * exp
          | Pair of exp * exp
          | Fst of exp
          | Snd of exp
          | Add of exp * exp
          | Sub of exp * exp
          | And of exp * exp
```

□