Symbols: s are alphanumeric strings that begin with a letter.

Terms: t are either a symbol or a pair.

Pairs: p are entered by writing (t1 . t2) where t1 and t2 are both terms.

Lists are either the symbol nil, which designates an empty list, or they are a pair where the second element of the pair is list.

The elements of a list are written following an open parenthesis. If the list is empty a close parenthesis is written, otherwise the first element of the pair is written followed by all the elements of the list held in the second element of the pair. Note: the first element of any pair may either be a symbol or a list.

Abstraction of a symbol from a term, produces a term:

[x]x => I

[x]y => K y

[x](t1 t2) => S [x]t1 [x]t2

Reduction of terms:

S t1 t2 t3 => t1 t3 (t2 t3)

K t1 t2 => t1

I t => t

Y h => h (Y h)

Expressions: e can be any term, or an abstraction

(Recursive) Functions: f are defined in terms of expressions.

def f = e => f = Y [f]e