**Assignment lambda**

**1. Define the functions “less than” and “greater than” of two numerical arguments.**

T = λ*xy.x*

F = λ*xy.y*

NOT = λ*x.x* F T

PRE = λ*nfx.n (*λ*gh.h (g f))* (λ*u.x*) (λ*u.u)*

ZERO = λ*x.x*FT

GT = λ*xy*.NOT (ZERO (SUB *x y*) )

LT = λ*xy*.GT *y x*

**2. Define the positive and negative integers using pairs of natural numbers**

CHECK = λ*n.*IF (ZERO ( PRE n) ) T F

**3. Define addition and subtraction of integers**

ADD = λ*abfx.af (bfx)*

SUB = λ*ab.b* PRE *a*

**4. Define the division of positive integers recursively**

IF = λ*fab.fab*

0 = λ*sz.z*

1 = λ*sz.s (z)*

DIVIDE = λ*ab*.IF (ZERO *b*) 0 IF (SUB *a b*) 1 (DIVIDE (SUB *a b*) *b*)

**5. Define the function n! = n \* (n - 1) … 1 recursively**

MUL = λ*xyz.x (yz)*

FACT = λ*n*.IF (ZERO *n*) 1 MUL (*n* (FACT (PRE *n*)) )

**6. Define the rational numbers as pairs of integers**

CONS = λ*pqf*.*(fp) q*

CAR *=* λ*p.p* T

CDR = λ*p.p* F

MAKE = λ*nd*.(CONS *n*) *d*

**7. Define functions for the addition, subtraction, multiplication and division of rationals.**

NUMERATOR = λ*x*.CAR *x*

DENOMINATOR = λ*x*.CDR *x*

RADD = λ*xy*.MAKE (ADD (

MUL ((NUMERATOR *x*) (DENOMINATOR *y*))

MUL ((NUMERATOR *y*) (DENOMINATOR *x*)) )

MUL ((DENOMINATOR *x*) (DENOMINATOR *y*)) )

RSUB = λ*xy*.MAKE (SUB (

MUL ((NUMERATOR *x*) (DENOMINATOR *y*))

MUL ((NUMERATOR *y*) (DENOMINATOR *x*)) )

MUL ((DENOMINATOR *x*) (DENOMINATOR *y*)) )

RADD = λ*xy*.MAKE (MUL ((NUMERATOR *x*) (NUMERATOR *y*))

MUL ((DENOMINATOR *x*) (DENOMINATOR *y*)) )

RDIV = λ*xy*.MAKE (MUL ((NUMERATOR *x*) (DENOMINATOR *y*))

MUL ((NUMERATOR *y*) (DENOMINATOR *x*)) )

**8. Define a data structure to represent a list of numbers.**

[] := λcn. n

[1, 2, 3] := λcn. c 1 (c 2 (c 3 n) )

**9. Define a function which extracts the first element from a list**

HEAD = λ*l.l* (λ*ab.a*)

**10. Define a recursive function which counts the number of elements in a list**

LENGTH = λ*l.* IF (null *l*) 0 ADD (1 (LENGTH *l*))