

Problem Set 2 - Solutions
COMP301 Fall 2020
Week 3: 19.10.2020 - 23.10.2020

P1. A.

- **Unary Representation:** Each number is represented with a list that contains that number many #t's.
- **BigNum Representation:** Each number is represented in a different base given. Each of the digits will be represented by a number and the entire number will be represented as a list.

P1. B.

- Codes for Unary Representation:

```
(define create
  (lambda (x)
    (cond
      ((eqv? x 0) '())
      (else
       (cons #t (create (- x 1)))))))

(define is-zero? (lambda (x) (null? x)))

(define successor (lambda (x) (cons #t x)))
```

- Codes for BigNum Representation:

```
(define create
  (lambda (x b)
    (cond
      ((eqv? x 0) '())
      (else
       (cons (remainder x b) (create (quotient x b) b))))))

(define is-zero? (lambda (expr) (eqv? expr '())))

(define successor
  (lambda (expr b)
    (cond
      ((< (+ (car expr) 1) b) (cons (+ (car expr) 1) (cdr expr)))
      (else
       (cons 0 (successor (cdr expr) b))))))
```

P1. C.

- **create:** Constructor. This function only creates a new representation with respect to the given number.
- **is-zero?:** Observer, Predicate. This procedure does not create a representation or does not return any argument from the representation given. It only returns a true or false validation value.
- **successor:** Constructor. This procedure again creates a new representation from the given input.

P2.

```
(define count-free-occurrences
  (lambda (var exp)
    (cond
      ((symbol? exp) (if (eqv? var exp) 1 0))
      ((eqv? (car exp) 'lambda)
       (if (eqv? var (car (cadr exp)))
           0
           (count-free-occurrences var (caddr exp))))
      (else
       (+
        (count-free-occurrences var (car exp))
        (count-free-occurrences var (cadr exp)))))))
```