**Structures and Interpretation of Computer Program**

**Exercise Chapter 1.2 Name:** Wan Huzaifah bin Wan Azhar

**Exercise 1.2.4 Exponentiation**

**1.16 Design a fast exponentiation procedure using iterative process**

(define (square b)

(\* b b))

(define (even? n)

(= (remainder n 2) 0))

(define (fast-exp b n)

(fast-exp-iter b n 1))

(define (fast-exp-iter b n a)

(if (= n 0) a

(if (even? n)

(fast-exp-iter (square b) (/ n 2) a)

(fast-exp-iter b (- n 1) (\* a b))

)))

(display (fast-exp 4 14100))

**1.17 Design a fast multiplication procedure with logarithmic steps using recursive process**

(define (double b)

(\* b 2))

(define (halve b)

(/ b 2))

(define (even? n)

(= (remainder n 2) 0))

(define (fast-mult a b )

(if (= b 0) b

(if (even? b)

(fast-mult (double a) (halve b))

(+ (fast-mult a (- b 1)) a)

)))

(display (fast-mult 30 324))

**1.18 Design a fast multiplication procedure with logarithmic steps using iterative process**

(define (double b)

(\* b 2))

(define (halve b)

(/ b 2))

(define (even? n)

(= (remainder n 2) 0))

(define (fast-mult a b)

(fast-mult-iter a b 0))

(define (fast-mult-iter a b c)

(if (= b 0) c

(if (even? b)

(fast-mult-iter (double a) (halve b) c)

(fast-mult-iter a (- b 1) (+ c a))

)))

(display (fast-mult 30 324))

**1.19 Design fast Fibonacci iterator**

P’ and Q’ can be calculated by applying a and b one times.

Calculation (omitted details):

a’ = qb + qa + pa = (p+q)a + bq

b’ = qa + pb

a” = (p+q)a’ + b’q = (p+q)(qb+qa+pa) + (qa+pb)q = … = a(2pq+2q2+p2) + b(2pq + q2)

bq from a’ applied to a’’ becomes b(2pq+q2), q’ = 2pq+q2

b” = qa’ + pb’ = q(qb+qa+pa) + p(qa+pb) = … = a(2pq + q2) + b(q2+p2)

pb from b’ becomes b(q2+p2) after b”, therefore p’ = q2+p2

(define (fib n)

(fib-iter 1 0 0 1 n))

(define (fib-iter a b p q count)

(cond ((= count 0) b)

((even? count)

(fib-iter a

b

(+ (expt p 2) (expt q 2))

(+ (\* 2 p q) (expt q 2))

(/ count 2)))

(else (fib-iter (+ (\* b q) (\* a q) (\* a p))

(+ (\* b p) (\* a q))

p

q

(- count 1)))))