CSC236 Assignment 4

1.)

Postcondition: Returns a^b.

P(n): If $a \in R$, $b \in N$, and $n = |ab|$ ($|ab|$ being the absolute value of ab), rec_exp(a, b) terminates and returns a^b.

By the precondition we have $a \in R$ and $b \in N$.

Case 1: $b = 0$.

Then the condition for the if-statement in line 1 is satisfied and the program goes into this block and line 2 is executed. So the program terminates at line 2.

Since $b = 0$, we know that $a^b = 1$, (because $x^0 = 1$ for all real numbers x). Then the postcondition holds since at line 2 the program returns 1.

Induction Step: Let $n = |ab|$. Suppose for all $j \in N$, $0 \leq j < n$, P(j) holds. [IH]

WTP: P(n) holds.

Case 2: $b \bmod 2 = 0$.

Then by line 4, x = rec_exp(a, b/2). $b/2 < b$, so $|ab/2| < |ab|$, so $0 \leq |ab/2| < n$. By the IH, rec_exp(a, b/2) terminates and returns $a^{b/2}$. So $x = a^{b/2}$. By line 5, the program terminates and returns $xx = a^{b/2}a^{b/2} = a^b$. Therefore rec_exp(a, b) terminates and returns a^b, as wanted.

Case 3: $b \neq 0$ and $b \bmod 2 \neq 0$.

Then by line 7, $x = \text{rec\_exp}(a, (b - 1)/2)$. $(b - 1)/2 < b$, so $|a(b - 1)/2| < |ab|$, so

$0 \leq |a(b - 1)/2| < n$. By the IH, rec_exp(a, (b - 1)/2) terminates and returns $a^{(b-1)/2}$. So $x = a^{(b-1)/2}$.

By line 8, the program terminates and returns $xxa = a^{(b-1)/2}a^{(b-1)/2}a = a^b$. Therefore rec_exp(a, b) terminates and returns a^b, as wanted.

Therefore P(n) holds and the proof is complete.

2.)

a)

LI(k): If the loop is executed at least k times, then

(i) $0 \leq r_k \leq a$.

(ii) $q_k = \sum_{j=1}^{(a - rk)/b} 1$.

b)

Suppose the precondition holds and the program terminates. Since the program terminates, the loop is exectued a finite number of times, say t.

By part (i) in LI, $0 \leq r_t \leq a$, and by the exit condition $r_t < b$, therefore $0 \leq r_t < b$.

By part (ii) in LI, $q_t = \sum_{j=1}^{(a - rt)/b} 1$, which means $q_t = 1(a - r_t)/b = (a - r_t)/b$.

So $a - r_t = bq_t$, $a = bq_t + r_t$.

Since $\sum^{(a - rt)/b}_{j = 1} 1 \geq 0$ and $q_t = \sum^{(a - rt)/b}_{j = 1} 1$, $q_t \geq 0$.

Therefore $a = bq + r$, $q \geq 0$ and $0 \leq r < b$.

c)

Suppose the precondition holds.

Suppose that $k \in N$ and the loop is executed at least $k + 1$ times, then by the loop exit condition, this means $r_k \geq b$. By part (i) in LI, $0 \leq r_{k+1} \leq a$ which means r is a natural number. By line 5, $r_{k+1} = r_k - b$ and since $b > 0$, $r_{k+1} < r_k$ and since every decreasing sequence of natural numbers is finite, the loop terminates. When the loop terminates the program executes line 6 where the program terminates.

3.)

P(n): If A is a list of tuples of integers, and $len(A) \geq 2$, and $e \in N$, $1 \leq e < len(A)$, and $n = e$, Find_Closest_Pair_Rec(A, e) terminates and returns a pair of points in A[0 : e + 1] with minimal distance.

By the precondition we have A is a list of tuples of integers, $len(A) \geq 2$, $e \in N$ and $1 \leq e < len(A)$.

Case 1: $e = 1$.

Then the condition for the if-statement in line 1 is satisfied, and the program goes into this block and line 2 is executed. So the program terminates at line 2.

Since $e = 1$, the only two points in A[0 : e + 1] = A[0 : 2] are A[0] and A[1] which means

(A[0], A[1]) is the pair of points in A[0 : e + 1] with minimal distance. Then the postcondition holds since at line 2 the program returns (A[0], A[1]).

Induction Step: Let n = e, n > 1. Suppose for all $j \in N$, $1 \leq j < n$, P(j) holds. [IH]

Case 2: $e \neq 1$.

Then by line 3, (p, q) = Find_Closest_Pair_Rec(A, e - 1). $e - 1 < e$ so $1 \leq e - 1 < n$. By the IH, Find_Closest_Pair_Rec(A, e - 1) returns a pair of points in A[0 : e] with minimal distance. So (p, q) is the pair of points in A[0 : e] with minimal distance.

By line 4, min = Distance(p, q) so min is the distance between points p and q.

By line 6, the program now enters a while loop.

LI(k): If the loop in lines 6 - 11 is executed at least k times, then

(i) $0 \leq i_k \leq e$.

(ii) $(p_k, q_k) = (A[e], A[i_k])$ if and only if $Distance(A[e], A[i_k]) < min_k$.

(iii) $min_k = Distance(p_k, q_k)$.

Suppose that $k \in N$ and the loop is executed at least k times, then by the loop exit condition, $i_{k-1} < e$ and by line 11, $i_k = i_{k-1} + 1$ until $i_k \geq e$ which means the loop terminates when $i_k = e$. Since the loop terminates, the loop is executed a finite number of times, say t. By part (i) in LI, $i_t \leq e$ and by the loop exit condition, $i_t \geq e$ so $i_t = e$. By part (ii) in LI, $(p_t, q_t) = (A[e], A[i_t])$ if and only if $Distance(A[e], A[i_t]) < min_t$, which means that for every k_{th} iteration of the loop, $Distance(A[e], A[i_k])$ is compared to min_k, and by part (iii) in LI, $min_k = Distance(p_k, q_k)$ and if $Distance(A[e], A[i_k]) < min_k$, then (p_k, q_k) is set to $(A[e], A[i_k])$ and if not, by line 3,

(p_k, q_k) = Find_Closest_Pair_Rec(A, e - 1). So by the t_{th} iteration, the distance between A[e] and every point in A[0 : e] has been checked to see if it is a pair of points, let's call this pair m, with distance less than Distance(Find_Closest_Pair_Rec(A, e - 1)) and if it is then (p_t, q_t) = m. I have already shown that Find_Closest_Pair_Rec(A, e - 1) is a pair of points in A[0 : e] with minimal distance and since merging A[0 : e] and A[e] together gives A[0 : e + 1], then by the t_{th} iteration, (p_t, q_t) is a pair of points in A[0 : e + 1] with minimal distance. Therefore the program terminates and returns (p, q) as wanted.