SECTION B

4. Let $wp(S, Q)$ denote the 'weakest precondition of S to Q'.

   a. Give a precise definition of $wp(S, Q)$.

   [6 marks]

   b. Let Q_1, Q_2 be arbitrary postconditions and let S be any program or routine. Which of the equivalences below is generally valid?

$$\begin{aligned} wp(S, \neg Q_1) &\leftrightarrow \neg wp(S, Q_1) \\ wp(S, Q_1 \wedge Q_2) &\leftrightarrow (wp(S, Q_1) \wedge wp(S, Q_2)) \\ wp(S, Q_1 \vee Q_2) &\leftrightarrow (wp(S, Q_1) \vee wp(S, Q_2)) \\ wp(S, Q_1 \rightarrow Q_2) &\leftrightarrow (wp(S, Q_1) \rightarrow wp(S, Q_2)). \end{aligned}$$

   In each case, if you believe the equivalence is valid, briefly justify your answer. If you believe the equivalence is not valid give a counter-example.

   [12 marks]

   c. Let S_1, S_2 be arbitrary routines, Q be any precondition and let $|$ denote non-deterministic choice. Find a formula equivalent to

$$wp(S_1 | S_2, Q)$$

   built up from the basic formulas $wp(S_1, Q)$ and $wp(S_2, Q)$ using the boolean connectives only.

   [5 marks]

   d. Calculate

   1. $wp(x := x + 1, x > 0)$
   2. $wp(x := y + 1; y := x - 1, x > 0 \wedge y < 0)$
   3. $wp(y := x - 1; x := y + 1, x > 0 \wedge y < 0)$
   4. $wp(x, y := y + 1, x - 1, x > 0 \wedge y < 0)$
   5. $wp(x := y; t := x; y := t, x = a \wedge y = b)$.

   [10 marks]

[Total 33 marks]

5. a. Define two mutually recursive procedures *Find_item_in_tree(Item, Tree)* and *Find_item_in_trees(Item, List_of_trees)*. The first procedure should return 'true' if and only if 'Tree' is a finite tree of type 'Item' and 'Item' labels some node in 'Tree'. The second procedure should return 'true' if and only if 'List_of_trees' is a finite list of finite trees of type 'Item' and 'Item' labels some node of some tree in the list 'List_of_trees'. You may define your procedures in pseudo-code, or in any sufficiently formal language.

[9 marks]

b. Define a suitable variant v for these two procedures and use it to prove that they both terminate.

[8 marks]

c. By induction on the variant v, prove that the two procedures you defined in part (a) always return the correct value.

[8 marks]

d. For each positive integer n, let Tree(n) = $(n, [])$, the one-node tree with the root node labelled by n. What would happen if you ran

Find_item_in_tree(3, (0, [Tree($2n$): $n > 0$])) ?

Explain your answer.

[8 marks]

[Total: 33 marks]

6. Consider the following function.

int function S(int $A[], st, end$)

{ $r := 0$

$i := st$

while $i < end$

{

$r := r + A[i]$

$i++$

}

return r

}

a. What does the function S do?

[4 marks]

b. Define suitable preconditions and postconditions for the function that guarantee that it will work.

[5 marks]

c. Define a loop variant and prove that the function terminates, if the precondtions from part (a) hold.

[6 marks]

d. Define a loop invariant and prove that the function meets the specifications given in part (a).

[7 marks]

e. Use this function S to define another function A that calculates the mean average value of a non-empty list of integers.

[5 marks]

f. Define suitable pre and postconditions for your function A and prove that A meets these specifications.

[6 marks]

[Total 33 marks]

TURN OVER

END OF PAPER