1.

2.

3.

CONTINUED

Section B

4. Consider the following mutually recursive programs.

$$\begin{array}{lcll} p1(0) & = & a; & \\ p2(0) & = & b; & \\ & & & \\ p1(n) & = & p1(n-1) + p2(n-1) & (n \geq 1); \\ p2(n) & = & p1(n) + p2(n-1) & (n \geq 1) \end{array}$$

where all variables and constants range over non-negative integers.

a.

By defining a suitable variant, prove that the procedures $p1(n)$ and $p2(n)$ both terminate for all non-negative integers n.

[10 marks]

b. Calculate the values of $p1(i)$ and $p2(i)$ for $i = 0, 1, 2, 3$.

[5 marks]

Let $f_n(a, b)$ be the n'th term in the Fibonacci sequence

$$a, b, a + b, a + 2b, \ldots$$

i.e. the sequence defined by $f_0(a, b) = a$, $f_1(a, b) = b$ and for $n \geq 2$, $f_n(a, b) = f_{n-1}(a, b) + f_{n-2}(a, b)$.

c. Find formulas $\phi_1(n), \phi_2(n)$ using the Fibonacci sequence, above, such that for all integers $n \geq 0$, $p1(n) = \phi_1(n)$ and $p2(n) = \phi_2(n)$.

[8 marks]

d. Prove that $p1(n) = \phi_1(n)$ and $p2(n) = \phi_2(n)$ holds for all non-negative integers n by a careful induction argument. [Hint: it makes a difference whether you prove the result for $p1(n + 1)$ before the result for $p2(n + 1)$ or the other way round.]

[10 marks]

[Total for question 4: 33 marks]

5. Let $sp(S, P)$ denote the 'strongest postcondition of S from P'.

   a. Give a precise definition of $sp(S, P)$.

   [10 marks]

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

   $$sp(S_1|S_2, P)$$

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

   [8 marks]

   c. Calculate

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

   [15 marks]

[Total for question 5: 33 marks]

6. Consider the following bubble sort algorithm.

procedure BS(var int: $A[\,]$; int: i, j)

{

for $k := j$ down to $i + 1$

  for $l := i$ up to $k - 1$

    if $A[l] > A[l+1]$ then Swap$(A, l, l+1)$

}

You may assume that the procedure 'Swap' meets its specifications which are:-

Preconditions: $i, j \in dom(A)$.

Postconditions: $P = A_+[i] = A_-[j] \wedge A_+[j] = A_-[i] \wedge \forall k, (k \in dom(A) \wedge k \neq i \wedge k \neq j) \rightarrow A_+[k] = A_-[k]$.

a. Write suitable pre- and postconditions for the procedure BS which are met by this algorithm.

[6 marks]

[Total for part (a): 6 marks]

b. By establishing a suitable loop invariant for the (inner) l-loop, or otherwise, prove that on exiting the l-loop the following postconditions hold:

- $A^l_+[i], A^l_+[i+1], \ldots, A^l_+[k]$ are a permutation of $A^l_-[i], A^l_-[i+1], \ldots, A^l_-[k]$,
- for all m with $k < m \leq j$ we have $A^l_+[m] = A^l_-[m]$ and
- for all m with $i \leq m \leq k$ we have $A^l_+[k] \geq A^l_+[m]$.

where $A^l_+[m]$ and $A^l_-[m]$ stand for the values of $A[m]$ on exit to the l-loop and on entry to the loop, respectively.

[13 marks]

[Total for part (b): 13 marks]

c. i. Define a suitable invariant for the (outer) k-loop.

[5 marks]

ii. Prove that the k-loop invariant you defined in part (c)(i) is true at every iteration of the k-loop.

[5 marks]

iii. Using your k-loop invariant prove that the procedure's postconditions (from part (a)) are met on exiting the k-loop.

[4 marks]

[Total for part (c): 14 marks]

[Total for question 6: 33 marks]

END OF PAPER