Worth: 3% **Due:** By 12 noon on Tuesday 13 March.

---

1. (a) Write a detailed argument that shows that the loop invariant holds just before the loop condition is evaluated for the first time, under the assumption that the precondition is true.

Note: the loop invariant should have referred to natural numbers that are less then len(A).
Let S be the set of natural numbers that are less than len(A).
Since the precondition says that len(A) > 0, S is non-empty.
Assume that the precondition holds:
    Then A is a non-empty list.
    Then just before the loop condition is evaluated for the first time,
      $x = A[0]$ and $i = 1$ (by the first two statements in the algorithm).
    Let $k_0 = 0$.
    Then $k_0 \in S$.
    Then $k_0 < i$. # since $i = 1$
    Then $A[k_0] = x$.
    Then $\exists k \in S, k < i \wedge A[k] = x$.
    Assume $j \in S \wedge j < i$
        Then $j = 0$.
        Then $A[j] \leqslant x$.
    Then $\forall j \in S, j < i \Rightarrow A[j] \leqslant x$
    Then $\exists k \in S, k < i \wedge A[k] = x$ and $\forall j \in S, j < i \Rightarrow A[j] \leqslant x$.
Then, if the precondition holds, the loop invariant also holds just before the loop condition is evaluated for the first time.

(b) Assuming that the loop invariant is correct, write a detailed argument that shows that the postcondition will be satisfied once the loop terminates.

We should start by translating the postcondition into a logical statement.
"x is the largest value found in the list A" means $\exists k \in S, A[k] = x \wedge \forall j \in S, A[j] \leqslant x$.
Assume that the loop invariant holds:
    Then $\exists k \in S, k < i \wedge A[k] = x$ and $\forall j \in S, j < i \Rightarrow A[j] \leqslant x$.
    Assume that the loop terminates, *i.e.*, the loop condition evaluates to False:
        Then $i \not< \text{len}(A)$.
        Then, $i = \text{len}(A)$ (because i is initially $\leqslant$ len(A), is $<$ len(A) at the start of each iteration of the loop, and is increased by 1 on each iteration of the loop.)
        Then since the loop invariant holds,
            $\exists k \in S, k < i \wedge A[k] = x$ and $\forall j \in S, j < i \Rightarrow A[j] \leqslant x$.
        Then $\exists k \in S, k < \text{len}(A) \wedge A[k] = x$ and $\forall j \in S, j < \text{len}(A) \Rightarrow A[j] \leqslant x$.
            # subst. $i = \text{len}(A)$
        Then $\exists k \in S, A[k] = x$ and $\forall j \in S, A[j] \leqslant x$.
    Then, once the loop terminates, the postcondition holds.
Then, if the loop invariant holds and the loop terminates, the postcondition holds.

(c) Write a detailed argument that shows that the loop invariant is correct. That is, show that the loop invariant is true each time the program evaluates the loop condition.

Notation: To simplify from the notation used in the course notes, we will use a "prime symbol" to represent the values of variables at the end of one iteration of the loop (when those values are different from the ones at the beginning of the iteration). For example, "i′" represents the value of i at the end of the iteration, while "i" represents the value at the beginning.
Assume the loop invariant is true at the start of the iteration

Then $\exists k \in S, \mathrm{k} < \mathrm{i} \wedge \mathrm{A[k]} = \mathrm{x}$ and $\forall \mathrm{j} \in S, \mathrm{j} < \mathrm{i} \Rightarrow \mathrm{A[j]} \leqslant \mathrm{x}$.

Then either $\mathrm{A[i]} \leqslant \mathrm{x}$ or $\mathrm{A[i]} > \mathrm{x}$.

Case 1:
Assume $\mathrm{A[i]} \leqslant \mathrm{x}$
Then $\mathrm{x}' = \mathrm{x}$. # x unchanged in loop
Then $\exists k \in S, \mathrm{k} \leqslant \mathrm{i} \wedge \mathrm{A[k]} = \mathrm{x}'$ and $\forall \mathrm{j} \in S, \mathrm{j} \leqslant \mathrm{i} \Rightarrow \mathrm{A[j]} \leqslant \mathrm{x}'$.
 # note: $<$ became $\leqslant$
Case 2:
Assume $\mathrm{A[i]} > \mathrm{x}$
Then $\mathrm{x}' = \mathrm{A[i]}$.
Then $\mathrm{x}' > \mathrm{x}$.
Then $\exists k \in S, \mathrm{k} \leqslant \mathrm{i} \wedge \mathrm{A[k]} = \mathrm{x}'$ and $\forall \mathrm{j} \in S, \mathrm{j} \leqslant \mathrm{i} \Rightarrow \mathrm{A[j]} \leqslant \mathrm{x}'$.
Then, in either case, $\exists k \in S, \mathrm{k} \leqslant \mathrm{i} \wedge \mathrm{A[k]} = \mathrm{x}'$ and $\forall \mathrm{j} \in S, \mathrm{j} \leqslant \mathrm{i} \Rightarrow \mathrm{A[j]} \leqslant \mathrm{x}'$.
Then $\exists k \in S, \mathrm{k} < \mathrm{i}' \wedge \mathrm{A[k]} = \mathrm{x}'$ and $\forall \mathrm{j} \in S, \mathrm{j} < \mathrm{i}' \Rightarrow \mathrm{A[j]} \leqslant \mathrm{x}'$.
 # $\mathrm{i}' = \mathrm{i} + 1$ means $a \leqslant \mathrm{i} \Rightarrow a < \mathrm{i}'$
Then the loop invariant is true at the end of the iteration.

Then the loop invariant is true at the start of the iteration $\Rightarrow$ that the loop invariant is true at the end of the iteration.
Then the loop invariant is true each time the program evaluates the loop condition.

(d) In order to prove that the algorithm is correct, we need to show that the loop eventually terminates. Provide an argument that shows that the loop condition is eventually false.

We are given that A is a non-empty list, so we know that len(A) is at least 1 and is also finite. The variable i is initially set to 1. If the list A has only one element in it, the loop condition will be false the first time it is checked. Otherwise, since i is incremented by 1 on each iteration of the loop, and since len(A) is finite, i will eventually be equal to len(A). In either case, the loop condition is eventually false.

2. (a) Write a detailed argument that shows that the loop invariant holds just before the loop condition is evaluated for the first time, under the assumption that the precondition is true.

Note: the problem referred to y as both a positive integer and a postive natural number. These sets are of course the same, but it would have been better to have used the same wording.

Assume that the precondition holds:
Then x is a natural number and y is a positive natural number.
Then just before the loop condition is evaluated for the first time,
$\mathrm{r} = \mathrm{x}$ (by the first statement in the algorithm).
Then $\mathrm{r} \geqslant 0$. # since $\mathrm{x} \in \mathbb{N}$ and $\mathrm{r} = \mathrm{x}$
Let $\mathrm{q}_0 = 0$.
Then $\mathrm{q}_0 \in \mathbb{N}$.
Then $\mathrm{x} = \mathrm{y}\mathrm{q}_0 + \mathrm{r}$. # since $\mathrm{x} = \mathrm{r}$ and $\mathrm{y}\mathrm{q}_0 = 0$.
Then $\exists q \in \mathbb{N}, \mathrm{x} = \mathrm{y}\mathrm{q} + \mathrm{r}$.
Then $\mathrm{r} \geqslant 0 \wedge \exists q \in \mathbb{N}, \mathrm{x} = \mathrm{y}\mathrm{q} + \mathrm{r}$.

Then, if the precondition holds, the loop invariant also holds just before the loop condition is evaluated for the first time.

(b) Assuming that the loop invariant is correct, write a detailed argument that shows that the postcondition will be satisfied once the loop terminates.

We should start by translating the postcondition into a logical statement.
"r = x%y for natural number x and positive natural number y " means $x \in \mathbb{N}, y \in \mathbb{N}^+, r \in \mathbb{N}, r < y \wedge \exists q \in \mathbb{N}, x = qy + r$.
Assume that the loop invariant holds:
  Then $r \geqslant 0 \wedge \exists q \in \mathbb{N}, x = yq + r$.
  Assume that the loop terminates, *i.e.*, the loop condition evaluates to False:
    Then $r \ngeqslant y$.
    Then, $r < y$
    Then since the loop invariant holds,
    $r \geqslant 0 \wedge r < y \wedge \exists q \in \mathbb{N}, x = qy + r$.
    Then since we are given that x is a natural number, y is a positive natural number, and r is formed from the subtraction of these quantities and $r \geqslant 0$,
    $x \in \mathbb{N}, y \in \mathbb{N}^+, r \in \mathbb{N}, r < y \wedge \exists q \in \mathbb{N}, x = qy + r$.
  Then, once the loop terminates, the postcondition holds.
Then, if the loop invariant holds and the loop terminates, the postcondition holds.

(c) Write a detailed argument that shows that the loop invariant is correct. That is, show that the loop invariant is true each time the program evaluates the loop condition.

Notation: As before, we use a "prime symbol" to represent the values of variables at the end of one iteration of the loop.
Assume the loop invariant is true at the start of the iteration

  Then $r \geqslant 0 \wedge \exists q \in \mathbb{N}, x = yq + r$
  Let $q_0 \in \mathbb{N}$ be such that $x = yq_0 + r$.
  Then $x = y(q_0 + 1) - y + r$.
  Let $q_1 = q_0 + 1$.
  Then $q_1 \in \mathbb{N}$.
  Then $x = yq_1 + (r - y)$.
  Then $r' = r - y$.   # line 3 of program
  Then $x = yq_1 + r'$.
  Then $r' \geqslant 0$.   # by the loop condition $r \geqslant y$ and definition of r'
  Then $r' \geqslant 0 \wedge \exists q \in \mathbb{N}, x = yq + r')$
  Then the loop invariant is true at the end of the iteration.

Then the loop invariant is true at the start of the iteration $\Rightarrow$ that the loop invariant is true at the end of the iteration.
Then the loop invariant is true each time the program evaluates the loop condition.

(d) In order to prove that the algorithm is correct, we need to show that the loop eventually terminates. Provide an argument that shows that the loop condition is eventually false.

We are given that $x \in \mathbb{N}$ and $y \in \mathbb{N}^+$. The variable r starts as a finite value. In the loop, since $y > 0$, the statement $r = r - y$ makes r smaller each iteration. Since $y \in \mathbb{N}^+$, y is finite, and so we must eventually have that the value of r is less than y. That is, we must eventually have that $r \ngeqslant y$. The loop must terminate because the loop condition is eventually false.