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

**Remember to write the the *full name* and *student number* of each member of your group prominently on your submission. Your submission must be a PDF file named e6.pdf and it must be handed-in using the MarkUs system. You may create the PDF file using a typesetting system (export to PDF) or by scanning in handwritten work to create a PDF file.**

Each exercise may be completed in groups of 1 – 2 students who are in the **same** tutorial section.

*Please read and understand the policy on Collaboration given on the Course Information Sheet. Then, to protect yourself, list on the front of your submission **every** source of information you used to complete this homework (other than your own lecture and tutorial notes, and materials available directly on the course webpage). For example, indicate clearly the **name** of every student with whom you had discussions, the **title** of every additional textbook you consulted, the **source** of every additional web document you used,* etc.

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions.

---

1. Consider the following python function:

```
def largest(A):
    '''Return the largest value found in the list A.'''

    # Precondition: A is a non-empty list (i.e., len(A) > 0)
    x = A[0]              # (line 1)
    i = 1                 # (line 2)

    # Loop Invariant: there exists a natural number k, A[k] == x and
    #                   for all natural numbers j, j < i implies A[j] <= x
    while i < len(A):     # (line 3)
        if A[i] > x:      # (line 4)
            x = A[i]      # (line 5)
        i = i + 1         # (line 6)

    # Postcondition: x is the largest value found in the list A.
    return x              # (line 7)
```

   (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.

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

   (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.

   (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.

**Question 2 is on the next page.**

2. Now consider the following python function:

```
def remainder(x,y):
    '''Return x % y for natural number x and positive integer y.'''

    # Precondition: x is a natural number, y is a positive natural number.
    r = x            # (line 1)

    # Loop Invariant: r >=0  and
    #                 there exists a natural number q such that x = yq + r
    while r >= y:    # (line 2)
        r = r - y    # (line 3)

    # Postcondition: r = x % y for natural number x and positive integer y.
    return r         # (line 4)
```

(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.

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

(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.

(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.