EECS 3101 Fall 2021 (B & E) – Assignment 1

Remember to write your **full name** and **student number** prominently on your submission. To avoid suspicions of plagiarism: at the beginning of your submission, **clearly state any resources (people, print, electronic) outside of the course materials, and the course staff, that you consulted**. You must submit a PDF file with the required filename. The PDF file could be generated using LaTeX(recommended), Microsoft Word (saved as PDF, **not** the `.docx` file), or other editor/tools. The submission must be **typed**. Handwritten submissions are **not** accepted.

Due October 1, 2021, 10:00 PM (on eClass); required file(s): a1sol.pdf

Answer each question completely, always justifying your claims and reasoning. Your solution will be graded not only on correctness, but also on clarity. Answers that are technically correct that are hard to understand will not receive full marks. Mark values for each question are contained in the [square brackets]. Your submitted file does **not** need to include/repeat the questions — just write your solutions.

The assignment must be completed individually.

1. **[10]** Consider the following program which sorts a list that only contains letters A, B or C. For example, the input list `[B, A, A, C, A, B]` is sorted into `[A, A, A, B, B, C]`, and the input `[A, C, A, C]` is sorted into `[A, A, C, C]`.

```
// Pre:  <lst> is non-empty and contains only 'A', 'B', or 'C'.
// Post: <lst> is sorted in non-descending order, alphabetically.
void SortABC(char[] lst) {
    int p1 = 0;
    int p2 = 0;
    int p3 = 0;
    while (p3 < lst.length) {
        if (lst[p3] == 'A') {
            swap lst[p1] and lst[p3];
            p1 += 1;
            p2 = max(p2, p1);
        }
        if (lst[p3] == 'B') {
            swap lst[p3] and lst[p2];
            p2 += 1;
        }
        p3 += 1;
    }
    return;
}
```

   (a) Find the proper loop invariant for the while-loop (`Inv(p1, p2, p3)`) so that it is sufficient for proving the correctness of `SortABC`.

   (b) Prove the invariant that your find in (a), using induction. **Hint**: The proof of the invariant could be divided into different cases of the processed part of the array — try to divide the cases in a simple way (i.e., keep the number of possible cases relatively small) and make sure to analyse all possible cases thoroughly.

   (c) Use the invariant to prove that the postcondition of `SortABC` is satisfied if the loop terminates.

   (d) Prove that the loop must terminate after a finite number of iterations.

2. [**10**] Consider the following program which returns the **least common multiple** (LCM) of two natural numbers a and b, i.e., the smallest positive integer $m > 0$ such that $m \bmod a = 0$ and $m \bmod b = 0$. For example, $\text{LCM}(4, 5) = 20$ and $\text{LCM}(4, 6) = 12$.

```
// Pre:  a, b are natural numbers, a >= 1, b >= 1
// Post: return LCM(a, b)
int GetLCM(int a, int b) {
    int x = a;
    int y = b;
    while (x != y) {
        if (x < y) {
            x = x + a;
        }
        else {
            y = y + b;
        }
    }
    return x;
}
```

(a) Find the proper loop invariant (`Inv(x, y)`) for the while-loop so that it is sufficient for proving the correctness of `GetLCM`. **Hint:** The invariant should probably include (but not limited to) the following conjuncts: $x \leq \text{LCM}(a, b)$ and $x > 0$. Also, your invariant should be as simple as possible given that it is sufficient for proving the correctness of the function.

(b) Prove the invariant that your find in (a), using induction.

(c) Use the invariant to prove that the postcondition of `GetLCM` is satisfied if the loop terminates.

Note: For this question, you are **not** required to prove that the loop terminates. However, you are encouraged to give it a shot. Try to find an expression (in terms of x and y) such that (i) the loop body is guaranteed to decrease the value of the expression in each iteration, and (ii) the expression must be ≥ 0 while in the loop. The combination of (i) and (ii) guarantees that the loop will terminate. (Why?)