CS411 Homework 3
Program Verification using Hoare Logic — *Due: 27 February 2007*

Corrections [last updated 2/21, 1pm]:
Feb 20: typo in Problem 2: "$n - 1$" instead of "$n + 1$" in the postcondition.
Feb 20: Problem 2 has been changed (the old question is optional)
Feb 19: typos has been fixed in the code in Problem 1.

Hand in the written part of the assignment next Wednesday in class, and submit the code using CMS. Please write your solutions very neatly.

1. Using the Hoare rules, prove the following partial correctness statement:

$$\{x = m \land y = n \land z = 1\}\ Pow\ \{z = m^n\}$$

where *Pow* is the following program:

```
while $\neg(y = 0)$ do (
    while $even(y)$ do (
        $x := x * x$;
        $y := y/2$
    );
    $z := z * x$;
    $y := y - 1$
)
```

Here, $y/2$ is the integer division of y by 2, and $even(y)$ means that y is an even number. [*Hint:* The two loops have the same invariant, which involves x^y as a subexpression.]

2. Weakest preconditions and verification conditions.

(a) Derive the weakest precondition $wp(c, y = 42)$ for the command c shown below. Simplify the computed precondition as much as possible.

```
if $(a \geq 0)$ then $b := b + 2$ else $b := b * a$;
if $(b \geq a)$ then $y := a * 7$ else $y := b - 3$
```

(b) Consider the following partial correctness statement:

$$\{n \geq 0\}\quad i := 0; s := 0; \text{while } (i \neq n) \text{ do } (s := s + i;\ i := i + 1)\quad \{2 * s = (n - 1) * n\}$$

Derive an appropriate loop invariant, and use the invariant to build a verification condition for the above correctness triple.

3. In this problem you are asked to use the ESCJava system to annotate the program `List.java` shown below with appropriate preconditions, postconditions, and loop invariants. Please read the documentation in the provided code.

Your code must successfully verify using ESCJava, and must show that the sorting code is correct. Each method must be given the approapriate specification, and each loop must be annotated with the appropriate loop invariant.

```
public class List {
    private int a[];
    private int n;

    public List(int m){
        a = new int[m];
        n = 0;
    }

    public void add(int element){
        a[n] = element;
        n++;
    }

    public void sort(){
        int i, j;
        for(i = 0; i < n-1; i++)
            for(j = i+1; j < n; j++)
                if (a[j] < a[i])
                    swap(i,j);
    }

    public void swap(int i, int j) {
        int t = a[i];
        a[i] = a[j];
        a[j] = t;
    }
}
```

To do: submit the annotated `List.java` file using CMS.

Optional Problems

The following are optional problems. Each of them will count for 5 additional (bonus) points.

1. [**Optional Written Problem**]. Suppose we tweak the Hoare rule for while loops as follows:

$$\frac{\{P \wedge b\}\ c\ \{P\}}{\{P\}\ \textsf{while}\ b\ \textsf{do}\ c\ \{P\}}$$

   Is the new rule sound? Explain.

2. [**Optional Programming Problem**]. Implement your favorite ADT (Abstract Data Type), such as a stack or a queue, and annotate it with the appropriate ESCJava specification. Submit a file `ADT.java` using CMS.