Remember these are the steps you must follow when proving an iterative program is correct:

- Clearly state the loop invariant “p” that you will use.
- Use induction to prove that p is a loop invariant. (So for the base step you prove that p is true just before the loop is first entered, and for the inductive step you prove \( p \land \text{cond}\{M_L\}p \) where \( M_L \) is the loop body and \( \text{cond} \) is the condition.)
- Now use your loop invariant to prove that the given program is partially correct with respect to the given initial and final assertions.
- Finally, prove the given program is correct with respect to the given initial and final assertions (i.e. show that it always terminate).

**Practice Exercises**

1. Use a loop invariant to prove the correctness of the following algorithm to compute the maximum of a sequence of integers \( a_0, \ldots, a_{n-1} \) with respect to the initial assertion \( n \geq 1 \) and the final assertion that \( \text{ans} = \max\{a_0, \ldots, a_{n-1}\} \).

```plaintext
procedure computeMax(\{a_0, \ldots, a_{n-1}\})
    \text{ans} = a_0
    i = 1
    while (i < n)
        if a_i > \text{ans} then \text{ans} = a_i;
        i = i + 1
    return \text{ans}
```

2. For the following program: (1) give a meaningful loop invariant \( p \), and (2) assuming that \( p \) is a loop invariant (you need not prove that for this problem), argue that the program is correct under the initial assertion that \( n \) is a positive integer.

```plaintext
procedure factorial(n)
    \text{ans} = 1
    i = 2
    while (i \leq n)
        \text{ans} = \text{ans} \ast i
        i = i + 1
    return \text{ans}
```

3. Prove that the loop invariant that you gave in practice problem 2 is a loop invariant. (For the quiz, I will write up the loop invariant for you if this problem is selected.)
4. For the following program: (1) give a meaningful loop invariant \( p \), and (2) assuming that
\( p \) is a loop invariant (you need not prove that for this problem) prove that the program
is correct under the initial assertion that \( n \) is a positive integer.

```
procedure mult(n, m)
    prod = 0
    k = n
    while (k > 0)
        k = k - 1
        prod = prod + m
    return prod
```

5. Prove that the loop invariant that you gave in practice problem 4 is a loop invariant.
(For the quiz, I will write up the loop invariant for you if this problem is selected.)

6. Consider the following buggy program to compute \( ans = n(n + 1)/2 \) with respect to the
initial assertion that “\( n \) is a positive integer” (i.e. \( n \geq 1 \)). It was designed using loop
invariant
\( p: \ "(ans = i(i + 1)/2 \land (i \leq n))". \) However, as written, \( p \) is not a loop invariant.
Modify the program so that \( p \) is truly a loop invariant and then prove it is correct.

```
procedure sum(n)
    ans = 0
    i = 0
    while (i \leq n - 1)
        ans = ans + i
        i = i + 1
    return ans
```

Problems to Submit

1. (15 pts) Prove the correctness of the following iterative program for finding the \( n \)th
Fibonacci number with respect to the initial assertion \( n \geq 0 \) and the final assertion that
\( y \) is the \( n \)th Fibonacci number, \( f_n \). Recall \( f_0 = 0 \), \( f_1 = 1 \) and \( \forall n \geq 2 \), \( f_n = f_{n-1} + f_{n-2} \).

```
procedure fibonacci(n)
    if (n==0) then y = 0
    else
        x = 0
        y = 1
        for i = 1 to n - 1
            z = x + y
            x = y
            y = z
        return y
```
2. (15 pts) Use a loop invariant to prove that the following program is correct with respect to the initial assertion “$a$ and $d$ are positive integers” and the final assertion “$q$ and $r$ are integers such that $a = dq + r$ and $0 \leq r < d$.

```plaintext
procedure divide(a, d)
  r = a
  q = 0
  while (r ≥ d)
    r = r - d
    q = q + 1
  return {q, r}
```

*Hint:* Think about why this works. It may help to trace the procedure on a few examples. What’s the relationship between $a, d, q, r$?

3. (15 pts) Use a loop invariant to prove that the following program is correct with respect to the initial assertion that $x$ is a positive integer and the final assertion that $ans = x^2$.

```plaintext
procedure square(x)
  i = j = 1
  while (i < x)
    j = j + 2i + 1
    i = i + 1
  return j
```

**Challenge Problem**

Use a loop invariant to prove that the following program is correct with respect to the initial assertion that $n$ is an integer $\geq 0$ and the final assertion that $\text{palindrome}(n)$ returns the integer obtained by reversing the digits in $n$. (For example, if $n = 1432$ then it should return $2341$.)

```plaintext
procedure palindrome(n)
  reverse = 0
  m = n
  while m > 0
    temp = m%10
    reverse = reverse * 10 + temp
    m = (m - temp)/10
  return reverse
```