Due, Tuesday September 30. Due in my mailbox before class, or at the beginning of class, no exceptions! No delay until 5pm because we will talk about the homework as part of an exam review.

Practice Problems:

1. 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 \( \text{ans} = x^2 \).

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

2. Use the idea of the “fast exponentiation” algorithm discussed in class to make a “faster multiplication” algorithm that returns the product of \((x,y)\). Your program should be asymptotically faster than one that repetively adds \( x \) to itself \( y \) times. (or \( y \) to itself \( x \) times). You are not allowed to multiply in your algorithm. You are permitted to divide by two. [as an aside, for binary numbers, this can be implemented as a simple bit shift, so it isn’t really cheating].

Problems to turn in:

1. Consider the game of robertNim. This game starts as a single pile of marbles. Each turn, players can take 1, 2, 3 or 5 marbles. (Note, this is not a typo, you can take, one, or two, or three, or five marbles). Whoever takes the last marble loses.

   (a) Make a chart and list the optimal move for player 1 for starting piles of size 1 through 11.

   (b) Define a proposition that states when player 1 has an optimal move. This proposition might have a form similar to: \( P(n) \) : “player 1 can will if the number of marbles in the pile can be expressed as \( 3n + 1 \) or \( 3n + 2 \).”

   (c) Use induction to prove your proposition correct.

2. Consider the following very small (one line!) Hoare triples. For each, say if the triples are valid. Argue for their correctness, including the proof that the final assertion is true and that the program terminates. If the Hoare triple is not correct, state why. You may assume that \( x \) is an integer.

   (a) \( \{ x > 0 \} \ x := x + 1 \ \{ x > 0 \} \)

   (b) \( \{ x > 0 \} \ x := x - 1 \ \{ x > 0 \} \)

   (c) \( \{ x < 0 \} \)
      while \( x \neq 0 \) do
         \( x := x + 1 \)
      od
      \( \{ x = 0 \} \)

   (d) \( \{ \text{true} \} \)
      while \( x \neq 0 \)
         \( x := x + 1 \)
      od
      \( \{ x = 0 \} \)