Computer Science 81, Spring 2010
Assignment 6
Due Tue. Mar. 2

1. Try a few of the simpler JAPE "Variable Programs" proofs for warm-up. You don't need to turn these in.

2. Prove the total correctness of the following program in the "Japeish" language, presented here as a Hoare triple. **All variables are integers**.

```
{n = n0 ∧ n0 ≥ 0 ∧ b > 0}

(s := b; r := 1)
while n > 0
      do
      if mod(n, 2) = 1
            then r := r × s
            else skip
      fi;
      n := n ÷ 2;
      s := s × s
      od

{r = pow(b, n0)}
```

Here ÷ is the integer division function (quotient is always an integer), *mod* is the modulus or remainder function, that is, mod(n, d) is the remainder of dividing n by d, and *pow* is the power function, that is pow(b, n0) = b^{n0}.

The first task is to determine for your self how this program works. Next devise an appropriate invariant and verify it informally. Then determine a variant for termination.

Finally, using your invariant and variant, enter the program with the invariant embedded into JAPE as a conjecture and use JAPE to edit the proof.

Below is the incantation that should be entered into JAPE as a conjecture, except you will have to insert your invariant, replace the various symbols with the ones from JAPE's symbol repertoire and this will all be on one line:

```
WHERE DISTINCT b, n, n0, pow, r, s IS
{n=n0 ∧ n0 ≥ 0 ∧ b>0} (s:=b; r:=1)
{… your invariant here …}
while n > 0 do if mod(n,2)=1 then r:=r×s else skip fi; n:=n÷2; s := s×s od
{r = pow(b, n0)}
```

Note that you would start the proof using JAPE's Ntuple rule since the invariant is present as an intermediate assertion.

Any facts that you use that are not built into JAPE's Hoare logic should be expressed either using Lemmas (the proofs of which can use the **Obviously** rule). List all lemmas that you used.

Here are most of the lemmas I used, and you are welcome to use these. You will need to enter them in the Useful Lemmas window, and provide "proofs", most of which you can declare as obvious for now. Generally, your proof will be more elegant with fewer lemmas.

(A, B, C, X are arbitrary)

```
a.  pow(A, B) defined
b.  mod(n,2) computes
c.  2 ≠ 0
d.  A×B×C = A×(B×C)
e.                A = X  |— 1×A = X
f.                B = 1  |— A×B = A
f.                B = C  |— A×B = A×C
h.         A = B, B = C  |— A = C
i.      A ≥ 0, ¬(A > 0)  |— A = 0
j.                n ≥ 0  |— n ÷ 2 ≥ 0
k.         n = A, n > 0  |— n ÷ 2 < A
l.                n = 0  |— pow(A, n) = 1
m.      ¬(mod(n,2) = 1)  |— mod(n, 2) = 0
n.         mod(n,2) = 0  |— pow(s × s, n ÷ 2) = pow(s, n)
o.         mod(n,2) = 1  |— s × pow(s × s, n ÷ 2) = pow(s, n)
```

Save your work often in JAPE, especially your lemmas. If you get an error dialog from JAPE asking you whether to Continue or Quit, it is better to Continue, because if you quit, you may lose your current proof.

To reload your work, load the Theory first, then load your file.

Submit: Your proof on paper, along with a commentary showing what the various parts are for, and how they inter-relate. This could be something along the lines of slides presented in class, or some other approach. For example, you could use the JAPE line numbers and have your commentary refer to ranges of them.

Note that you might have to repeat the proof one or more times, so you can see how to construct the commentary. It should go faster the second time. You might want to make notes along the way, because the rationale and order is not always evident in the final proof.

For reference, my proof was about 87 lines of JAPE, not counting the proofs lemmas. If you need more hints, please ask.