Homework 6

This homework is due Monday, Mar. 19 (by 5:00 PM). For problems where you are to draw the run-time stack or a display, please draw them so that they grow towards the top of the page.

1. Consider the following Pascal program.

```
program problem1;
  var x: integer;
  var b: boolean;

  procedure foo(x: integer);
    var y: integer;

    procedure bar(z: integer);
      begin
        {** draw stack when this comment is reached}
      end {bar};

    begin
     bar(x)
    end {foo};

  procedure snafu(q: integer);
    var s,t: integer;
    begin
      s := 1;
      foo(q+s);
    end {snafu};

  begin
    snafu(3);
  end.
```

Draw the run-time stack, and any auxillary registers, when execution reaches the comment in `bar` using:

(a) (6 points) static chaining, being sure to clearly show both the static and dynamic links, and the SP and EP registers;

(b) (6 points) a display, being sure to show the display, the saved display elements on the stack, the SP register, and the dynamic links.

2. Consider the following Pascal program.

```
program problem2(input,output);
  var nine: real;

  function average(a,b: real): real;
    begin
      {** draw stack when this comment is reached}
      average := (a+b) / 2.0;
    end {average};

  function mysqrt(x: real): real;

    function improve(guess: real): real;
      begin
        improve := average(guess, x / guess)
      end {improve};

    function good(guess: real): boolean;
      begin
        good := abs(guess*guess - x) < 1.0
      end {good};

    function sqrt_iter(guess: real): real;
      begin
        if good(guess) then sqrt_iter := guess
                         else sqrt_iter := sqrt_iter(improve(guess))
      end {sqrt_iter};

    begin
     mysqrt := sqrt_iter(1.0)
    end {mysqrt};

  begin {main part of problem2 program}
    nine := 9.0;
    writeln(mysqrt(nine));
  end.
```

The above program makes the calls to `sqrt_iter` with arguments: 1.0, 5.0, 3.4, and 3.0235, and then these return with 3.0235 as the final answer.

Draw the run-time stack, and any auxillary registers, when execution reaches the comment in `average` during the third recursive call to `sqrt_iter`, when `a` is 3.4 (i.e, the third execution of `average`) using:

(a) (10 points) static chaining, being sure to clearly show both the static and dynamic links, and the SP and EP registers;

(b) (10 points) a display, being sure to show the display, the saved display elements on the stack, the SP register, and the dynamic links.

3. (5 points) Do exercise 6-7, which starts on page 241 in MacLennan.
4. (10 points) Do exercise 5 on page 258 in MacLennan (procedure call in an Algol-like language vs. in FORTRAN). Recall that in FORTRAN, there is no recursion.