For all problems in problem set 2:

1) Solutions to **Part 2** must be submitted both electronically (Malia Kilpinen, malia@mit.edu) and on paper. Email problem sets with a title in the following format: LastName_PSXx.ad[bs]. Due date: **10/7/02, 3pm.**
2) All submitted problems must contain your name, email address, and which problem you are submitting.
3) The following should be written as comments in the code: a short description of what the program is doing, and also document what the most important variables, data structures, and constants are, and how they are used in the program.
4) In addition to 3) each module should have a "header comment" with the following information:

```
------------------
-- Module name: Name of module
-- Explanation: What is the purpose of this module. Explain it so a person
--              unfamiliar with the code can understand.
-- Input: Input to this module from the calling unit.
-- Output: Data that is returned to calling unit (for procedures).
-- Return value: Value that is returned to calling unit (for functions).
-- Comments: Extra information that might be needed for the unit to execute properly.
------------------
```

Problem set 2: Procedures and functions

Part 1

Write a program that communicates with the user via a menu. The user can to choose between different operations, e.g., add two numbers; calculate N!, and so on. To solve this problem you have to write a number of *procedures* and *functions* to do the calculations and other required operations. You have to use parameters to send data to and from the procedures and functions. The functions do not have any output parameters though.

The Main routine is given and looks as follows:

```
--  Something …
procedure Some_Math_functions is
-- this is where your routines go
begin
  Menu;
end Some_Math_functions;
```

16.35 – problem set 2 – 9/30/02

Problem 1: function N_Factorial

Write a function to compute the factorial of N (where 'N' is an integer). The function must be generic i.e. it must compute the factorial for any integer 'N'. Use the template shown below:

```
---------------------
-- Module: N_Factorial
-- Explanation: Calculates N! i.e., 1*2*3*…*N
-- Input: A positive Integer (N)
-- Return value: Factorial of the input value (N!)
-- Comments: N has to be a positive and if calculating N!
--           generates an overflow, this has to be taken
--           care of.
---------------------
function N_Factorial (N : in Integer) return Integer is

begin
  -- your code
end N_Factorial
```

Problem 2: function Max_Number

Write a function that accepts two floating point numbers and that returns the larger of the two.

```
---------------------
-- Module: Max_Number
-- Explanation: Finds the larger of two floating point
                numbers. If both numbers are identical, one
                of them will be returned.
-- Input: Two floating point numbers
-- Return value: The larger of the two numbers
-- Comments: State which of the two equal numbers you would
--           select and why.
---------------------
```

Problem 3: procedure Add_Integers

Write a procedure that adds two integers (submitted to the procedure via two parameters) and return the sum via the third parameter of the procedure.

```
---------------------
-- Module: Add_Integers
-- Explanation: Add two integers and return the sum
-- Input: Two integers
-- Return value: The sum (an integer)
-- Comments: How do you account for overflow.
---------------------
```

Problem 4: procedure Menu
Write a procedure that handles a menu.

```
---------------------
-- Module: Menu
-- Explanation: Handles a menu where one can choose between
                adding integers, find the larger of two
                floats or calculate N! for an N. All input
                from the terminal shall be handled by this
                procedure (or subunits)

                The menu should also have an alternative to
                exit the program, and return to Main.
-- Input: --
-- Return value: --
-- Comments: --
---------------------
```

Part 2

Split your program into a number of separate parts. The parts that should be moved are the functions N_Factorial, Max_Number, and the procedure Add_Integers. These should be included in your program, but placed in separate files. The rest of the procedures does not have to be moved from your main file.

The goal behind this exercise is to get you comfortable with working with multiple source files.