1. (28 pts) Using the C++ programming language, indicate the binding time (language design, language implementation, compilation, link, run, etc.) for each of the following attributes. Justify your answer.

   (a) The variable declaration that corresponds to a certain variable reference (use)
   (b) The range of possible values for integer numbers
   (c) The meaning of char
   (d) The address of a local variable
   (e) The address of a library function
   (f) The referencing environment of a function passed as a parameter
   (g) The total amount of memory needed by the data in a program

2. (24 pts) Can a language that uses dynamic scoping do type checking at compile time? Why? Can a language that uses static scoping do type checking at run time? Why?

3. (24 pts) Does Scheme use static or dynamic scoping? Write a short Scheme program that proves your answer.

4. (24 pts) Consider the following pseudo-code:

   ```plaintext
   x : integer;     -- global
   procedure set_x (n : integer)
       x := n;
   procedure print_x
       write_integer (x);
   procedure foo (S, P : procedure; n : integer)
       x : integer;
       if n in {1,3}
           set_x(n);
       else
           S(n);
       if n in {1,2}
           print_x;
       else
           P;
   ```
Assume that the language uses dynamic scoping. What does this program print if the language uses shallow binding? Why? What does it print with deep binding? Why?

*Note:* At exactly one point during execution in the deep binding case, the program will attempt to print an uninitialized variable. Simply write a “?” for the value printed at that point.