B Exam January 1996: Programming Languages

Problem 1: Types Explain the sentence "Programming language X is statically typed."

Is C statically typed? Is Scheme statically typed? If you respond positively, argue why the statement is true. If not, explain how the respective language violates a condition of *your* general definition.

Problem 2: Memory Safety Different languages offer different levels of safety with respect to memory access.

Explain why Scheme (or SML) programs cannot cause segmentation faults with memory accesses through vectors.

Explain why C programs often cause segmentation faults long *after* a mis-reference to array memory occurred. Why is it impossible to fix this problem transparently in extensions of C? (Equivalently: why do good C++ libraries provide separate, programmer-defined vector classes?)

Your explanations should be complete but not verbose.

Problem 3: Parameter Passing Ada's general design document specified that implementors of compliers could choose among call-by-value-result and call-by-reference as parameter-passing mechanisms for certain procedure parameters.

1. What would be the output of the following pseudo-program under a call-by-reference regime for `f`? And what would be the output if `f` were implemented with call-by-value-result?

```
define int x = 0;

define procedure f (z)
  set z to z+1;
  return (x = z)
end;

print f(x);
```

2. Briefly explain how to implement the two techniques for scalar variables.

3. Explain one negative implication of the decision of Ada's designers to leave the interpretation of parameter-passing up to the implementors of individual compilers.