The course project is to build a compiler for a small language. This is a "living" document will be revised throughout the semester until it is a complete, if sometimes informal, language specification. Revisions may include additions, removals and changes to meet pedagogical goals and to ensure internal consistency.

(X) means zero or one occurrence of X { X } + means one or more occurrences of X { X } * means zero or more occurrences of X

SECTION 1: Lexical structure (see version 1.1 of this document)
SECTION 2: Syntactic structure (see version 2.0 of this document)
SECTION 3: Type checking and semantics (see version 3.0 of this document)
SECTION 4: Intermediate code generation (see version 3.0 of this document)
SECTION 5: Assembly code generation (this document)
SECTION 6: Machine-independent optimizations (version 5.0 of this document)

At this stage of the project the main goal is to get the compiler to generate assembly language instructions that preserve the semantics of the original source program. The ISA you are generating for is called SmallPond. Documentation will be distributed later this week.

The expectation is that your compiler will:

1) generate assembly code that preserves semantics of source code program
2) perform appropriate register allocation and assignment

Use the various sample programs you have written up to this point to help you test the correctness of your code generation.

Small updates/clarifications:

ARRAY ACCESS
Accessing the size of each array dimension: if a is an n-dimensional array, allow a to used in a record access construct as well: a._0 through a._(n-1) give access to the sizes of each of the n dimensions. Taking a concrete example, if foo is a 3-dimensional array of character and we reserve foo(5,3,7), then foo._0 has value 5, foo._1 has value 3, and foo._2 has value 7.

Note that no modification of the grammar is needed. Instead, your type-checking simply needs to permit the name of an array to used with the '.' record operator and the special labels (which are already valid identifiers) of the form _k, where k is a constant, and which is between 0 and n-1, inclusive, for an n-dimensional array.

POINTER SIZE
The size of a pointer: 32 bits (4 bytes)

RESERVE & IMPLICIT ASSIGNMENT
reserve X not only reserves an appropriate amount of memory to store a value of the type of X, but stores the pointer to that block of memory in the variable X.

SUBMISSION & GRADING:
Submit your code using Autolab. Submissions are due no later than 5:00 PM on Monday April 23.