CSE 3302 Lab Assignment 3

Due October 16, 2013

Goals:

1. Understanding of Pascal.
2. Understanding of scope.

Requirements:

1. Extend the solution to Lab 1 from Spring 2013 to give warnings about names that are re-declared ("masked"):

   a. If the new declaration is at the same level of scope, indicate that the new declaration is RISKY.
   b. If the new declaration is at a nested level of scope, indicate that the new declaration is acceptable.

   For either warning, include the *code index* in the program listing for the masked declaration, along with a precise "^" indication of the offending name. (Optional: You may either replace or include line numbers in the program listing, along with referencing the line numbers in the warnings.)

2. Email your code, as an attachment, to `sourabh.bose@mavs.uta.edu` by 12:45 p.m. on October 16, 2013.

Getting Started:

1. The code to modify is at `http://ranger.uta.edu/~weems/NOTES3302/LAB1SPR13/plzero.io.pas`

2. You may assume your input is a syntactically-correct PL/0 program, but with questionable scoping practices to be detected.

3. A few details ordered by their appearance in the PL/0 environment source code:

   a. The global `cx` is valuable (see `gen`).
   b. `table` will need to have `level` for all identifiers. (Wirth had a severe memory constraint.) Originally, `constant` did not need a `level`, since the value was compiled into the `lit` instruction.
   c. You will steal from the formatting in `error`.
   d. The parameter `lev` for `block` is golden.
   e. `enter` is where the action is . . . (Flames for all . . .)
   f. `in` and `out` are at a scope level above the source code being compiled. It is not RISKY to mask these.

4. To get started, you might add code to `block` to selectively print the contents of `table`.

```
-bash-3.2$ cat test1.pl0 - | plzero.io.dm
    0 const a=5;
    1
    1 var i,i;
 ****         ^ masks declaration at 1 RISKY, same scope
    1
    1 procedure a;
 ****            ^ masks declaration at 0 RISKY, same scope
    1
    1 const i=20;
 ****            ^ masks declaration at 1 acceptable, nested
    2 var a,in;
 ****        ^ masks declaration at 1 acceptable, nested
 ****           ^ masks declaration at -1 acceptable, nested
    2
    2   procedure i;
 ****               ^ masks declaration at 1 RISKY, same scope
    2
    2     const i=11;
 ****                ^ masks declaration at 2 acceptable, nested
    3     begin out:=i end;
    7
    7 begin
    8   out:=a
    9 end;
   11
   11 call a.
 start pl/0
! 0
 end pl/0
```