```
% Scanner for assignment 3
% TDDD08 Logic Programming
%
% top predicate:
% scan(+String, -Tokens)
%
% try: scan("x:=3; y:=1; while x>1 do y := y*x; x := x-1 od",Tokens).
%
% NOTE: strings are lists of ASCII codes, i.e.
% "Prolog" = [80,114,111,108,111,103]

scan([],[]).
scan([C|Cs],[';'|Ts]) :-
    semicolon(C),!,
    scan(Cs,Ts).
scan([C|Cs],Ts) :-
    space(C),!,
    scan(Cs,Ts).
scan([C|Cs],[num(T)|Ts]) :-
    digit(C),!,
    scan_number(Cs,Cs1,CNum),
    name(T,[C|CNum]),
    scan(Cs1,Ts).
scan([C1,C2|Cs],[T|Ts]) :-
    name(T,[C1,C2]),
    operator(T),!,
    scan(Cs,Ts).
scan([C|Cs],[T|Ts]):-
    name(T,[C]),
    operator(T),!,
    scan(Cs,Ts).
scan([C|Cs],[T|Ts]) :-
    letter(C),
    scan_key_or_id(Cs,Cs1,CWord),
    name(Word,[C|CWord]),
    classify(Word,T),
    scan(Cs1,Ts).

% scaning a number
% scan_number(+In, -Out, -Num)
% Num is a string of digits from front of In,
% Out is the remaining string

scan_number([C|Cs],Cs1,[C|CN]) :-
    digit(C),!,
    scan_number(Cs,Cs1,CN).
scan_number(Cs,Cs,[]).

% scaning a keyword or an identifier
% scan_key_or_id(+In, -Out, -Word)
% Word is a string from front of In,
% Out is the remaining string

scan_key_or_id([C|Cs],Cs1,[C|CW]) :-
    (letter(C)
     ;
     digit(C)
    ),!,
    scan_key_or_id(Cs,Cs1,CW).
scan_key_or_id(Cs,Cs,[]).

% distinguishing keywords from identifiers

classify(W,T) :-
    keyword(W),!,
    T = W.
classify(W,id(W)).


digit(C) :-
    C >= "0", C =< "9".
```

```
letter(C) :-
    C >= "a", C =< "z"
    ;
    C >= "A", C =< "Z".


semicolon(59).


operator('*').
operator('+').
operator('/').
operator('-').
operator('>').
operator('<').
operator('=').
operator('=<').
operator('>=').
operator(':=').


space(32).


keyword(skip).
keyword(if).
keyword(then).
keyword(else).
keyword(fi).
keyword(while).
keyword(do).
keyword(od).
keyword(true).
keyword(false).
```