The University of Western Australia
Dept. of Electrical & Electronic Engineering
A/Prof Thomas Bräunl

Intelligent Robotics
ENGT4311

Lab Assignment - Syntax&Semantics

Lab Assignment I1 Individual Assignment Due: week 3

Consider the programming language "*Drive*", defined in EBNF:

```
program     =   { statement  ";" } .
statement   =   TIMES constant DO program END |
                IF psd "<" constant DO program END |
                STRAIGHT | BACK | LEFT | RIGHT  .
psd         =   PSD_FRONT | PSD_LEFT | PSD_RIGHT .
constant    =   digit {digit } .          (* max. 5 *)
digit       =   '0' | .. | '9' .
```

Sample "Drive" program:

```
TIMES 4 DO STRAIGHT; RIGHT; END;
LEFT; LEFT;
IF PSD_FRONT < 40 DO STRAIGHT; END;
```

Part 1: Implement a scanner for the language symbols and constants in Drive
Print the input program as a sequence of tokens.

Part 2: Implement a recursive descent parser for Drive
The parser calls the scanner and checks the syntactical correctness of the program.
Error are to be reported – parsing stops at the first error encountered.

Note: White space (" ", "\n", "\r", "\t") may be contained between successive tokens.

Bonus point: Extend the language to handle variables.

Hints

1. Implement a function to return the current input symbol `symboltype CurSymbol();`

2. Use scanner from previous lab to advance one symbol `symboltype NextSymbol();`

3. Use function "error":

```
void error(char *s)
{ printf("ERROR: %s\n", s); /* print error message */
  myexit();                 /* terminate program */
}
```

4. Use function "check" to see if a specific keyword is present:

```
bool check(symboltype s)
{ if (s == CurSymbol())
  { NextSymbol(); return TRUE; }      /* symbol is present */
  else return FALSE;                  /* symbol is not there */
}
```

5. Use function "accept" if a specific keyword has to follow:

```
void accept(symboltype s)
{ if (!check(s)) error("wrong symbol");  /* must accept */
}
```