# Computer Science and Engineering
# IIIT Kalyani, West Bengal

## Compilers Design Laboratory (Spring: 2017 - 2018)

*3rd Year CSE: 6th Semester*

*Assignment - 3* *Marks:* 15

Assignment Out: *19th January, 2018* Report on or before: 28th *January, 2018*

1. Consider a *context-free grammar G*. Its *terminals* are $\Sigma = \{ + - * / = < > ( ) \{ \} := ;$ `and else end` *ic id* `if int do` *fc* `float not or print prog scan` *str* `then while` $\}$.

2. Key words are: `and else end if int do float not or print prog scan then while`.

3. *ic* stands for an unsigned integer constant, *id* is a C-like identifier, *fc* is a floating point constant, *str* is a string such as `"IIIT Kalyani"`.

4. Most of the other terminals are usual, where := is for assignment e.g. `a := b + 2*c` and = is the relational operator 'equal' etc.

5. Write a C function (without using Lex/Flex) `int yylex()` that will return a *token* corresponding to each terminal. A few *global variables* should hold attribute and other values related to a token: the actual text (lexeme) is available in `yytext[]`, the attribute value is available in `yylval`, line number in `yyline` and starting character position at `yypos`.

6. *Key-words* are reserved with specific meaning. Store them in a fixed array of character strings along with the corresponding *tokens*. While reading from the input, read both an identifier and a key word in a similar manner. Search the *key-word* in the table, if there is a match, return the appropriate *token*; otherwise return the token of an identifier.

7. The name of the C++ scanner file should be `myLex.c++` with the header file `myLex.h`. The program should print the stream of token in human understandable form. As an example the (token, value) stream for

   ```
   prog
       int a, b;
       scan a
       b := a + 5;
       print b
   end
   ```

   is

   ```
   <514,"prog"> <507,"int"> <505, "a"> <44,","> <505, "b">
   <59, ";"> <515, "sacn> <505, "a"> <505, "b"> <500, ":=">
   <505, "a"> <43, "+"> <504, "5"> <59, ";"> <513, "print">
   <505, "b"> <503, "end">
   ```

8. Prepare a `.tar` file

   ```
   $ tar cvf <RollNo ass3>.tar myLex.c myLex.h
   ```

   Send us the `.tar` file.