Lab 1

Tuesday, February 2, 2021 2:01 PM

Lex:

- .l extension
- Lex.yy.c program file is created everytime an input is given

Input: LEX source pgm
Output: C program automatically generated by Lex

Lex source code:

Declarations
%%
Translation rules
%%
Auxilliary procedures (C functions)

Declarations: C code:

```
%{
#include<stdio.h>
%}
```

Variables, regular definitions that can be used in the translation rules for pattern matching
User definable symbols

*: 0 or more
+: 1 or more

Translation rules:
P {action 1}

P: regular expression
Action: associated action when it matches the lexeme (sequence of characters in the input, variable, procedure) in the input

Auxialliary procedures:
Not necesssary, very big procedure? Can be added here

- LEX matches **longest matching pattern**
- If 2 rules are matched, it takes first rule
- If no match, copies input to output

```
$vi mypgm.l
$lex mypgm.l
$cc lex.yy.c -ll -o mypgm
//-ll => -libl.a
$./mypgm
//./filename (for ubuntu) or filename
OR
$mypgm < infile.c
//input file, < input read direction
$mypgm < infile. > outfile
//output file > output write direction
$more outfile
$cat outfile
```

Works as an interpreter
Takes one input at a time if no input file given
Ctrl+C to exit
Input is taken as character, convert to int/float as required
Exit vi with esc, ":wq!"

Source => LEX => yylex() in lex.yy.c

Input => yylex => output

Commenting:
```
/* here is the comment */
atof() //ascii to float
```

Programs done:
1. Identifier + if + while
2. Calculator