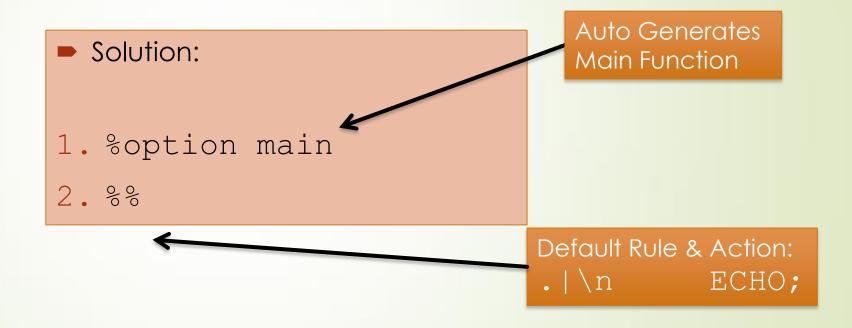
### Compiler Course Lab 2

**CSE 440** 

Task 1: Write a smallest possible Flex Program [Lex1.I] that compiles and works correctly

Hint: it should ECHO everything!

Task 1: Write a smallest possible Flex Program [Lex1.I] that compiles and works correctly



#define ECHO (void) fwrite( yytext, yyleng, 1, yyout )

## Task 2: Write a Flex Program [Lex2.l] that will print line no # and the line

Hint:

%option yylineno

Sets option that flex will start counting lines and store it in a global variable named yylineno

### Task 2: Write a Flex Program [Lex2.l] that will print line no # and the line

```
Solution:
1. %option yylineno
2. %option main
3. %%
4. ^\n printf("%4d\n", yylineno-1);
5. (.*) printf("%4d\t%s", yylineno, yytext);
6. \n ECHO;
7. %%
```

Task 3: Write a Flex Program [Lex3.I] that will do the same as Task 2 but it will take input from file and write output to another file, file names should be set from command line

- Hint:
- 1. FILE \*yyin is the file which by default flex reads from.
- 2. FILE \*yyout is the file to which ECHO actions are done.
- 3. These are global variables and can be reassigned by the user.

#### Task 3: Solution

```
1. %option yylineno
2. %option noyywrap
3. %%
4. ^\n fprintf(yyout, "%4d\n", yylineno-1);
5. (.*) fprintf(yyout, "%4d\t%s", yylineno, yytext);
6. \n ECHO;
7. %%
8. int main(int argc, char *argv[]) {
9.
  yyin = fopen(argv[1], "r");
    yyout = fopen(argv[2], "w");
10.
11.
    yylex();
12.
   fclose(yyin);
13. fclose(yyout);
14.}
```

Task 4: Write a Flex Program [Lex4.I] that will print line no # and occurrence of 'A','B','C' on the line

```
Hint:
1. % {
        int countA=0, countB=0, countC=0;
3. %}
                                        Section where you
4. %option yylineno
                                        can declare your own
                                        global variables and
5. %%
                                        include other files
6. ...
```

### Task 4: Solution [only rules part is shown]

```
1. ^\n
          fprintf(yyout, "%4d\n", yylineno-1);
2. a
        countA++;
3. b countB++;
4. c countC++;
5. .
6. \n {fprintf(yyout, "%4d\tA:%4d\tB:%4d\tC:%4d\n",
  yylineno-1, countA, countB, countC); countA=0;
  countB=0; countC=0; }
7. <<EOF>>> {fprintf(yyout, "%4d\tA:%4d\tB:%4d\tC:%4d",
  yylineno, countA, countB, countC); yyterminate();}
```

Detects End Of File

# Task 5: Write a Flex Program [Lex5.1] for following patterns and outputs

Outputs
Do nothing
Print <b>ID:lexeme</b>
Print <b>KEY:</b> lexeme
Print INT:integer_value
Print FLT:floating_value_in_decimal notation
Print <b>OP:lexeme</b>
Print NOT_RECOGNIZED

```
Hint: Use Regular Definition

DIGIT [0-9]
%%
{DIGIT}+ printf("INT");
```