Compiler Course Lab 2

CSE 440

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

- Hint: it should ECHO everything!

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

- Solution:

```
1. %option main
2. %%
```

Auto Generates Main Function

Default Rule & Action:

```
.|\n        ECHO;
```

```
#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:

```
%option yylineno
%option main
%%
^\n     printf("%4d\n", yylineno-1);
(.*)    printf("%4d\t%s", yylineno, yytext);
\n   ECHO;
%%
```

Task 3: Write a Flex Program [Lex3.l] 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

```
%option yylineno
%option noyywrap
%%
^\n       fprintf(yyout,"%4d\n", yylineno-1);
(.*)     fprintf(yyout,"%4d\t%s", yylineno, yytext);
\n    ECHO;
%%
int main(int argc, char *argv[]) {
   yyin = fopen(argv[1], "r");
   yyout = fopen(argv[2], "w");
    yylex();
    fclose(yyin);
   fclose(yyout);
}
```

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

- Hint:

```
%{
      int countA=0,countB=0,countC=0;
%}
%option yylineno
%%
…
…
```

Section where you can declare your own global variables and include other files

Task 4: Solution [only rules part is shown]

```
^\n     fprintf(yyout,"%4d\n", yylineno-1);
a       countA++;
b       countB++;
c       countC++;
.       ;
\n     {fprintf(yyout,"%4d\tA:%4d\tB:%4d\tC:%4d\n",
yylineno-1,countA,countB,countC); countA=0;
countB=0; countC=0;}
<<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.l] for following patterns and outputs

Patterns	Outputs
Blank Space, tab space, new line	Do nothing
C identifier	Print **ID:*lexeme***
if/else/switch/case/while/for	Print **KEY:*lexeme***
Any integer number	Print **INT:*integer_value***
Any float/double number	Print **FLT:*floating_value_in_decimal notation***
Any operator (+,-,*,/)	Print **OP:*lexeme***
Anything else	Print NOT_RECOGNIZED

Hint: Use Regular Definition

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