21AIE313 Introduction to Modern Compiler Design
S6 BTech AI

Lab Sheet 2

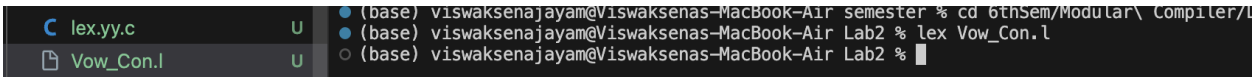
Name: J Viswaksena **Roll.No:** AM.EN.U4AIE21035

1. (a) Do the following lex program to count the number of Vowels and consonants from a given input. Save the lex file as VowCon.l

```
%{
#include<stdio.h>
int vowels=0, consonants=0;
%}
%%
[aeiouAEIOU] {vowels++;}
[A-Za-z]  {consonants++;}
%%
int main( int argc, char **argv )
{
       ++argv, --argc; /* skip over program name */
       if ( argc > 0 )
           yyin = fopen( argv[0], "r" );
       else
           yyin = stdin;
       yylex();
   printf ("Number of vowels=&d\n", vowels) ;
   printf("Number of consonants=%d\n", consonants) ;
}
int yywrap ( )
{
   return 1;
}
```

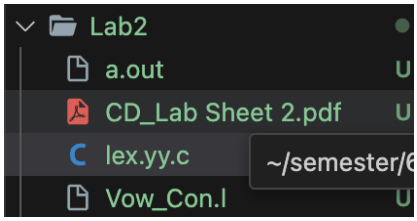
(b) Compile the lex file using the following command.

Once the lex file is compiled using the above code, a c file named lex.yy.c will be created.



(c) Compile the c file using the following code.
The executable file a.out will be created.

```
(base) viswaksenajayam@Viswaksenas-MacBook-Air Lab2 % cc lex.yy.c
```



(d) Run the file using the following command.

6thSem > Modular Compiler > Lab2 > input.txt

```
Vowals and
consonants
```

```
(base) viswaksenajayam@Viswaksenas-MacBook-Air Lab2 % ./a.out input.txt

Number of vowels = 6
Number of consonants = 13
```

2. Do the following lex program to implement a lexical analyzer that identifies various types of tokens such as identifiers, keywords, numbers, symbols, operators, and strings.

Code:

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

letter [a-zA-Z]
digit [0-9]

%%
"int"|"char"|"float"|"if"|"printf"    { printf("%s is a keyword\n", yytext); }
"+"|"-"|"*"|"/"|"^"|"="|"++"          { printf("%s is an operator\n", yytext); }
","|";"|"{"|"}"|"("|")"               { printf("%s is a special character\n", yytext);
}
```

```
{digit}+                                { printf("%s is a number\n", yytext); }
{letter}({letter}|{digit})*             { printf("%s is an identifier\n", yytext); }
[ \t\n]+                                /* ignore whitespace */
\"[^\"\n]*\"                            { printf("%s is a string\n", yytext); } /*
identify strings */
.                                       { printf("Error\n"); } /* ignore other
characters */
%%

int main(int argc, char **argv) {
   ++argv, --argc; /* skip over program name */
   if (argc > 0)
       yyin = fopen(argv[0], "r");
   else
       yyin = stdin;
   yylex();
}

int yywrap() {
   return 1;
}
```

Input.txt:

6thSem > Modular Compiler > Lab2 > q2 > input.txt

```
int main() {
    int a = 10;
    printf(a);
    return 0;
}
```

Commands and output:

```
(base) viswaksenajayam@Viswaksenas-MacBook-Air q2 % lex q2.l
(base) viswaksenajayam@Viswaksenas-MacBook-Air q2 % cc lex.yy.c
(base) viswaksenajayam@Viswaksenas-MacBook-Air q2 % ./a.out input.txt
int is a keyword
main is an identifier
( is a special character
) is a special character
{ is a special character
int is a keyword
a is an identifier
= is an operator
10 is a number
; is a special character
printf is a keyword
( is a special character
a is an identifier
) is a special character
; is a special character
return is an identifier
0 is a number
; is a special character
} is a special character
```

3. Do the following lex program to display the number of lines, words, and characters in an input text.

Code:

```
%{
#include<stdio.h>
int no_lines = 0, no_words = 0, no_chars = 0, no_other_char = 0, totalchar = 0;
%}

%%
\n      { no_lines++; no_words++; }
[\t ' ']    no_words++;
[A-Za-z0-9]    no_chars++;
.       no_other_char++;
%%

int main(int argc, char **argv) {
   ++argv; --argc; /* skip over program name */
   if (argc > 0)
       yyin = fopen(argv[0], "r");
   else
       yyin = stdin;

   yylex();

   totalchar = no_chars + no_other_char;

   printf("\n------ Result ------\n");
   printf("Number of lines: %d\n", no_lines);
   printf("Number of words: %d\n", no_words);
   printf("Number of alphanumeric characters: %d\n", no_chars);
   printf("Number of other characters: %d\n", no_other_char);
   printf("Total number of characters: %d\n", totalchar);
}

int yywrap() {
   return 1;
}
```

Input.txt

```
6thSem > Modular Compiler > Lab2 > q3 > input.txt
This is a sample input text.
It contains multiple lines.
Some words have numbers like 123 and others have symbols like @#$.
```

Commands and output:

```
(base) viswaksenajayam@Viswaksenas-MacBook-Air q3 % lex q3.l
(base) viswaksenajayam@Viswaksenas-MacBook-Air q3 % cc lex.yy.c
(base) viswaksenajayam@Viswaksenas-MacBook-Air q3 % ./a.out input.txt

------ Result ------
Number of lines: 3
Number of words: 22
Number of alphanumeric characters: 96
Number of other characters: 6
Total number of characters: 102
```