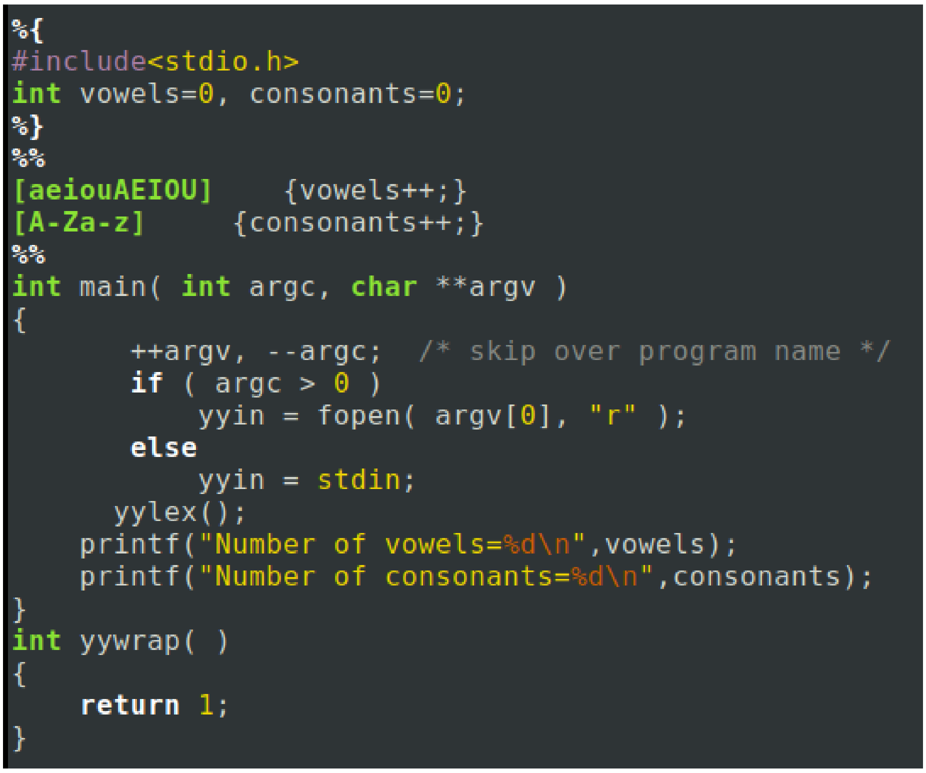
21AIE313 Introduction to Modern Compiler Design
S6 BTech AI

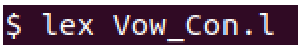
Lab Sheet 2

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;
}
```

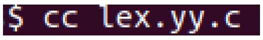
(b) Compile the lex file using the following command.



```
$ lex Vow_Con.l
```

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

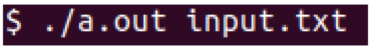
(c) Comiple the c file using the following code.



```
$ cc lex.yy.c
```

The executable file a.out will be created.

(d) Run the file using the following command.



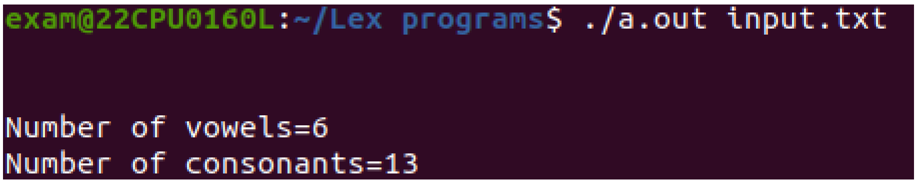
```
$ ./a.out input.txt
```

Sample input

input.txt

Vowals and
consonants

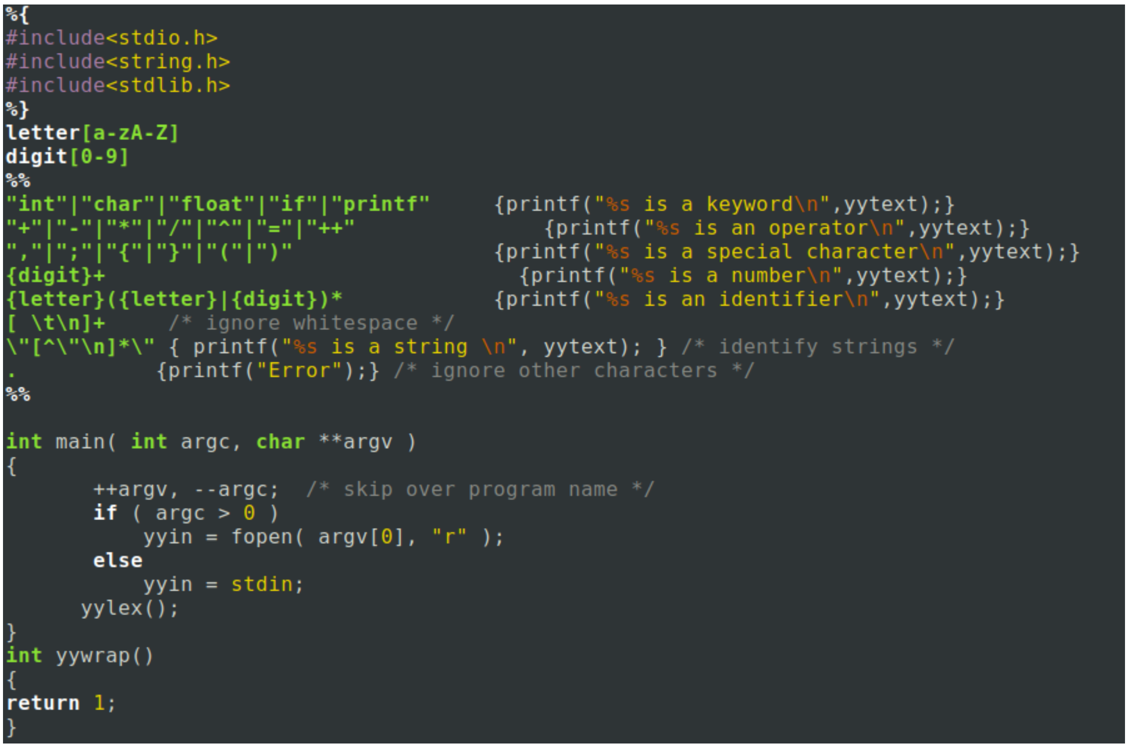
Sample Output



```
exam@22CPU0160L:~/Lex programs$ ./a.out input.txt


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

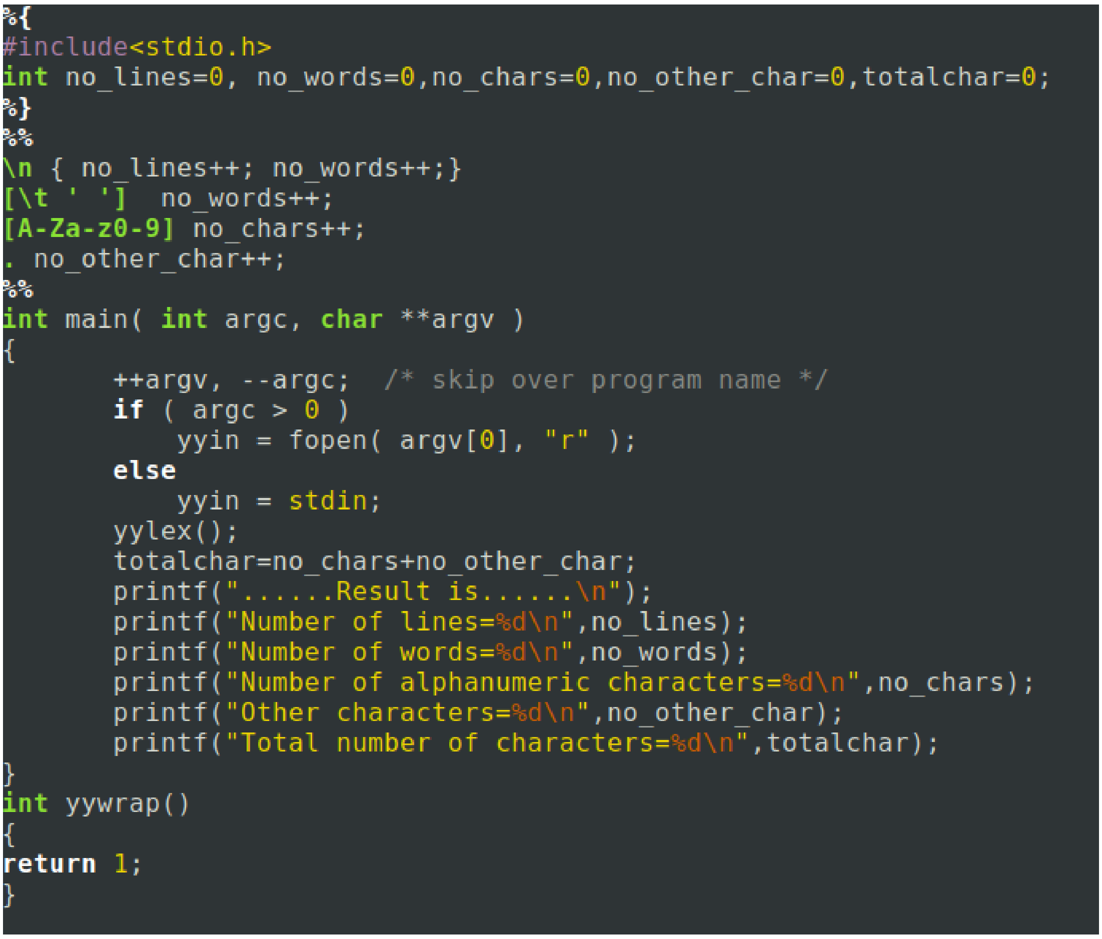
2. Do the following lex program to implement a lexical analyzer that identifies various types of tokens such as identifiers, keywords, numbers, symbols, operators, and strings.



```
%{
#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");} /* 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;
}
```

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



```
%{
#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("......Result is......\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("Other characters=%d\n",no_other_char);
       printf("Total number of characters=%d\n",totalchar);
}
int yywrap()
{
return 1;
}
```