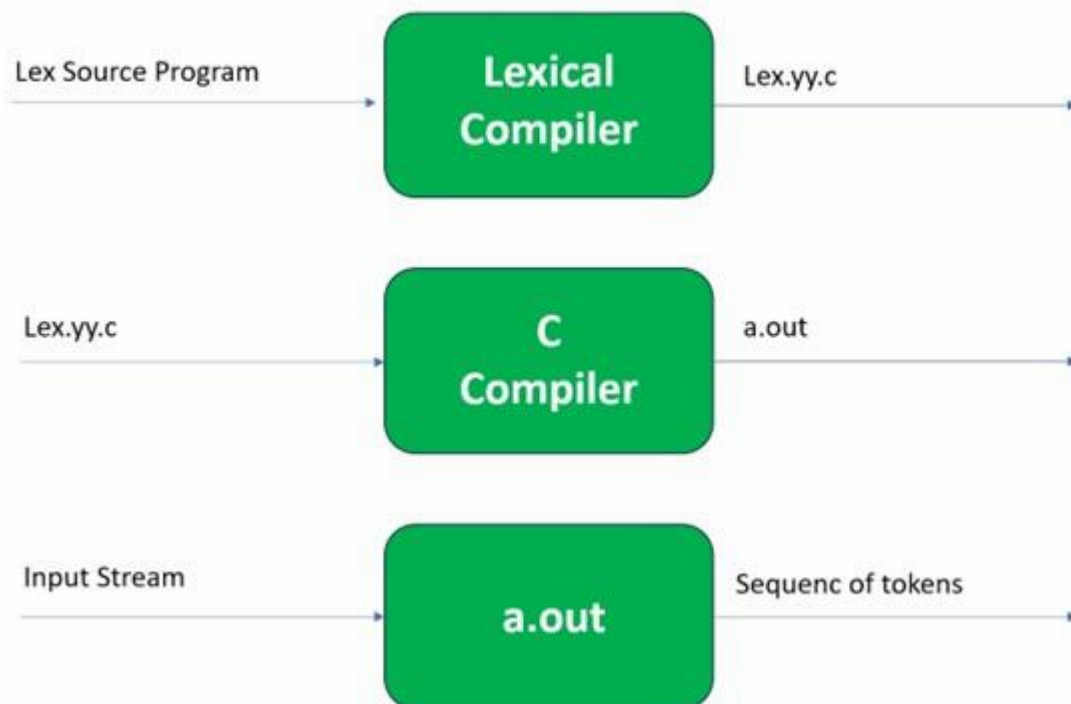


LAB 3

1. Write a lex program to find if given letter is vowel or consonants.
2. Write a lex program to count the number of vowels or consonants in a given string.
3. Write a lex program to check whether a given number is even or odd.

Theory:

Lex is a tool or a computer program that generates Lexical Analyzers (converts the stream of characters into tokens). The lex compiler takes the input and transforms that input into input patterns.



Here first source code is lex language with file name with extension '.l' is given as input to Lex compiler to get output as lex.yy.c.

Then lex.yy.c is used as input to the Compiler which gives the output in the form of a.out and finally a.out will take the streams of character and generate tokens as output.

Lex File Format

Lex program consists of three parts and is separated by %% delimiters.

Declarations

%%

Translation rules

%%

Auxiliary Procedures

Declarations: It includes declarations of variables.

Transition rules: These rules consist of Pattern and Action.

Auxiliary Procedures: The auxiliary section holds auxiliary functions used in the actions.

1. Write a lex program to find if given letter is vowel or consonants.

```
%{
#include <stdio.h>
%}
%%
[a|e|i|o|u|A|E|I|O|U]      printf("%s is a vowel\n", yytext);
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z]      printf("%s is a consonant\n", yytext);
[^a-zA-Z]      printf("%s is not a valid letter\n", yytext);
%%
int main() {
    printf("Enter a character: ");
    yylex();
    return 0;
}

int yywrap() {
```

```
return 1;
}
```

2. Write a lex program to count the number of vowels or consonants in a given string.

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

int vowels = 0;    // Counter for vowels
int consonants = 0; // Counter for consonants
}%

%%
[a|e|i|o|u|A|E|I|O|U]    { vowels++; }
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] { consonants++; }
. /* Ignore other characters */
%%

int main() {
    printf("Enter a string: ");
    yylex(); // Call lexical analyzer

    // Print the counts
    printf("\nNumber of vowels: %d\n", vowels);
    printf("Number of consonants: %d\n", consonants);

    return 0;
```

```
}
```

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

3. Write a lex program to check whether a given number is even or odd.

```
%{  
#include <stdio.h>  
%}  
  
%%  
[0-9]+ {  
    int num = atoi(yytext); // Convert matched text to integer  
    if (num % 2 == 0)  
        printf("%d is even\n", num);  
    else  
        printf("%d is odd\n", num);  
}  
.  
{  
    printf("Invalid input: %s\n", yytext);  
}  
%%  
  
int main() {
```

```
printf("Enter a number: ");  
yylex(); // Call lexical analyzer  
return 0;  
}
```

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