Experiment – 1
Name: Velchuri Sarvan
ID: CH.EN.U4CSE22160

**Aim:**

To compile the lex programs using the lex compiler and execute it using GCC compiler

1. Program to count the number of vowels and consonants in a given string.

2. Program to count the number of characters, words, spaces, end of lines in a given input file.

3. Program to count no of:
a) +ve and –ve integers
b) +ve and –ve fractions

4. Program to count the no of comment line in a given C program. Also eliminate them and copy that program into separate file

5. Program to count the no of ‘scanf' and ‘printf' statements in a C program. Replace them with ‘readf' and ‘writef' statements respectively.

**Algorithm and procedure:**

1. Save the code with .l extension.
2. Run the code ( lex <filename>.l ).
3. This will create an output file with the extension .yy.c which can be executed using the gcc compiler
4. Now run the command ( gcc <filename>.yy.c ).
5. This will generate an exe file as usual
6. Run the exe file with the command ( ./a.out  <by default> ) and the program will be executed successfully in the output terminal

Programs:

1.

```
%{
#include <stdio.h>
int vowels=0;
int cons=0;
%}
%%
[aeiouAEIOU] { vowels++; }
[b-df-hj-np-tv-zB-DF-HJ-NP-TV-Z] { cons++; }
%%
int yywrap()
{
    return 1;
}
int main()
{
    printf("Enter the string.. at end press ^d\n");
    yylex();
    printf("No of vowels=%d\nNo of consonants=%d\n", vowels, cons);
    return 0;
}
```

2.

```
%{
#include <stdio.h>
#include <stdlib.h>
extern FILE *yyin;
int c=0, w=0, s=0, l=0;
%}

WORD    [^ \t\n,\.:]+
EOL     [\n]
BLANK   [ \t]

%%

{WORD}  { w++; c += yyleng; }
{BLANK} { s++; c++; }
{EOL}   { l++; c++; }
.       { c++; }
%%
int yywrap()
{
    return 1;
}

int main(int argc, char *argv[])
{
    if(argc != 2)
    {
        printf("Usage: %s <input_file>\n", argv[0]);
        exit(1);
    }

    yyin = fopen(argv[1], "r");
    if(!yyin)
    {
        perror("Error opening file");
        exit(1);
    }
    yylex();
    printf("No of characters = %d\nNo of words = %d\nNo of spaces = %d\nNo of lines = %d\n", c, w, s, l);
    fclose(yyin);
    return 0;
}
```

3.

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

int posint = 0, negint = 0, posfraction = 0, negfraction = 0;
%}

%%
[-][0-9]+           { negint++; }
[+]?[0-9]+          { posint++; }
[+]?[0-9]*\.[0-9]+ { posfraction++; }
[-][0-9]*\.[0-9]+  { negfraction++; }
%%

int yywrap() { return 1; }

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <input_file>\n", argv[0]);
        exit(0);
    }
    FILE *yyin = fopen(argv[1], "r");
    if (!yyin) {
        perror("Failed to open input file");
        exit(1);
    }
    yylex();
    fclose(yyin);

    printf("No of +ve integers = %d\n", posint);
    printf("No of -ve integers = %d\n", negint);
    printf("No of +ve fractions = %d\n", posfraction);
    printf("No of -ve fractions = %d\n", negfraction);

    return 0;
}
```

4.

```
#include <stdio.h>
#include <stdlib.h>
int com = 0;
%}
%s COMMENT
%%
"/*"            { BEGIN(COMMENT); com++; }   // Count opening comment line
<COMMENT>"*/"   { BEGIN(INITIAL); }
<COMMENT>\n     { com++; }                    // Count every newline inside comment
<COMMENT>.      { /* consume other comment chars */ }

.|\n            { fprintf(yyout, "%s", yytext); }  // Copy other content

%%

int yywrap() {
    return 1;
}

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("Usage: %s <input_file> <output_file>\n", argv[0]);
        exit(0);
    }
    FILE *in = fopen(argv[1], "r");
    if (!in) {
        perror("Error opening input file");
        exit(1);
    }
    FILE *out = fopen(argv[2], "w");
    if (!out) {
        perror("Error opening output file");
        fclose(in);
        exit(1);
    }
    yyin = in;
    yyout = out;
    yylex();
    printf("No of comment lines = %d\n", com);
    fclose(in);
    fclose(out);
    return 0;
}
```

5)

```
%{
#include <stdio.h>
#include <stdlib.h>
int pc=0, sc=0;
%}
%%
“printf" { fprintf(yyout,"writef"); pc++;}
“scanf" { fprintf(yyout,"readf"); sc++;}
%%
int yywrap()
{
return 1;
}

int main(int argc, char *argv[])
{
if(argc!=2)
{
printf("Usage: <./a.out> \n");
exit(0);
}
yyin=fopen(argv[1],"r");
yyout=fopen(argv[2],"w");
yylex();
printf("No of printf statements = %d\n No of scanf statements=%d\n", pc, sc);
return 0;
}

```

**Output:**

1.

```
asecomputerlab@asecomputerlab:~$ flex vowel_count.l
flex: can't open vowel_count.l
asecomputerlab@asecomputerlab:~$ gedit vowel_count.l
asecomputerlab@asecomputerlab:~$ gedit vowel_count.l
asecomputerlab@asecomputerlab:~$ flex vowel_count.l
asecomputerlab@asecomputerlab:~$ gcc lex.yy.c -lfl -o vowel_count
asecomputerlab@asecomputerlab:~$ ./vowel_count
Enter the string.. at end press Ctrl+D (EOF)
Lex is powerful

No of vowels = 5
No of consonants = 8
asecomputerlab@asecomputerlab:~$
```

2.

```
asecomputerlab@asecomputerlab:~$ ./count_cwsl text.txt
No of characters = 40
No of words = 7
No of spaces = 4
No of lines = 3
```

3.

```
asecomputerlab@asecomputerlab:~$ ./count_number input.txt
echo "42 -99 +12.5 -0.33 0 100 -0.001" > test2.txt
./count_number test2.txt
echo "       " > test.txt
./count_number test.txt
No of +ve integers = 5
No of -ve integers = 1
No of +ve fractions = 1
No of -ve fractions = 2
```

4.

```
asecomputerlab@asecomputerlab-HP-ProDesk-400-G7-Microtower-PC:~/Desktop$ cat inp
ut.c
#include <stdio.h>

int main() {
    int a = 10; // not a block comment

    /* This is a
       multi-line comment */

    printf("Hello World\n");

    /* Single-line comment */

    return 0;
}
```