

COMPILER LAB REPORT

NAME – ANKUR MANNA

CLASS – BCSE III

ROLL NO : 18

SECTION – A1

ASSIGNMENT NUMBER 1

DEADLINE: 27TH March, 2021

SUBMITTED ON: 27TH March, 2021

REPORT SUBMITTED ON: 27TH March, 2021

1. Write a lex file to count the number of lines, words, and characters in the input.

```
%{
#include<stdio.h>
#include<string.h>
int i = 0;
int ch=0;
int ln=0;
}%

/* Rules Section*/
%%

([a-zA-Z0-9])* {i++;} /* Rule for counting number of words*/

. {ch++;}

"\n" {ln++; }
%%

int yywrap(void){}
```

```

int main(int argc ,char* argv[])
{
    if(argc > 1)
    {
        FILE *fp = fopen(argv[1], "r");
        if(fp)
            yyin = fp;
    }
    yylex();

    printf("%d %d %d",i,ch,ln);

    return 0;
}

```

OUTPUT:

```

ankur@DESKTOP-U0COJJT:/mnt/g/6th_sem/Compiler/lab$ lex p1.1
ankur@DESKTOP-U0COJJT:/mnt/g/6th_sem/Compiler/lab$ gcc lex.yy.c
ankur@DESKTOP-U0COJJT:/mnt/g/6th_sem/Compiler/lab$ ./a.out Tut.txt
56 66 3ankur@DESKTOP-U0COJJT:/mnt/g/6th_sem/Compiler/lab$ |

```

Input File Tut.txt

LEX is a tool used to generate a lexical analyzer. This document is a tutorial for the use of LEX for Expl
 Compiler development. Technically, LEX translates a set of regular expression specifications (given as input
 in input_file.1) into a C implementation of a corresponding finite state machine (lex.yy.c). This C program,

2. Write a lex file to count the number of numbers appearing in the input. Count the number of integers (without a decimal) separately from the number of floating point numbers (with a decimal, and at least one digit on either side of the decimal).

```
%{  
  
#include<stdio.h>  
  
/* Definition section */  
int integer=0;  
  
int fractions=0;  
  
%}  
  
/* Rule Section */  
DIGIT [0-9]  
%%  
  
\-?{DIGIT}+      {integer++;}  
  
\-?{DIGIT}*\. {fractions++;}  
  
"\n"    {  
    printf("\nNo. of Integrals: %d", integer);  
    integer = 0;  
    printf("\nNo. of fractions: %d", fractions);  
    fractions = 0;  
}  
  
%%  
  
// driver code  
int main()  
{  
    yylex();  
}
```

```
    return 0;
}
```

OUTPUT:

```
ankur@DESKTOP-U0COJTT:/mnt/g/6th_sem/Compiler/lab$ ./a.out
1 2.4 3.5 .2 5
```

```
No. of positive numbers: 2
```

```
No. of Positive numbers in fractions: 3|
```

3. Write a lex file to count the number of words in an input text that start with a vowel.

```
%{
    int vow_count=0;
    int const_count =0;
}%

ALPHA [a-zA-Z0-9]
%%

[aeiouAEIOU][A-Za-z]+ {vow_count++;}

[aeiouAEIOU]           {vow_count++;}

[A-Za-z(^aeiouAEIOU)][A-Za-z]+ {const_count++;}

"\n" {printf("%d\n", vow_count);  vow_count = 0;}
%%
int yywrap(){}
int main()
{
    printf("Enter the string of vowels and consonants:");
    yylex();
}
```

```
    return 0;  
}
```

OUTPUT:

```
ankur@DESKTOP-U0COJTT:/mnt/g/6th_sem/Compiler/lab$ lex p3.1  
ankur@DESKTOP-U0COJTT:/mnt/g/6th_sem/Compiler/lab$ gcc lex.yy.c  
ankur@DESKTOP-U0COJTT:/mnt/g/6th_sem/Compiler/lab$ ./a.out  
Enter the string of vowels and consonents:ankur ram shyam ankan  
    2  
ram shyam  
    0  
|
```

-----THE END-----