

Practical 2: Implement following programs using Lex

GAHAN SARAIYA (18MCEC10)

18mcec10@nirmauni.ac.in

I. AIM

1. Write a Lex program to print out all numbers from the given file.
2. Write a Lex program to printout all HTML tags in file.
3. Write a Lex program which adds line numbers to the given file and displaythe same onto the standard output.
4. Write a Lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file

II. A. WRITE A LEX PROGRAM TO PRINT OUT ALL NUMBERS FROM THE GIVEN FILE.

I. Lex file

```
%{  
#include <stdio.h>  
int num=0;  
%}  
%%  
[0-9] num++;  
%%  
int main(void) {  
yyin= fopen("f.in","r");  
yylex();  
printf("\nTotal Numbers in the files: %d\n",num);  
}  
int yywrap(void) {  
return 1;  
}
```

II. Input file

```
Howdy Doctor.  
call to 596-325-152  
Apartment No. 259
```

```
Sherlock lives in 221B backer street, London
```

III. Output

```
Howdy Doctor.  
call to --  
Apartment No.  
  
Sherlock lives in B backer street, London  
Total Numbers in the files: 15
```

III. B. WRITE A LEX PROGRAM TO PRINTOUT ALL HTML TAGS IN FILE.

I. Lex file

```
%{  
    #include <stdio.h>  
}%  
  
%%  
[0-9]* fprintf(yyout, "%s\n", yytext);  
%%  
  
int yywrap() {  
    return 1;  
}  
  
int main() {  
    // yyin=fopen("index.html", "r");  
    yyout=fopen("output.txt", "w");  
    yylex();  
    return 0;  
}
```

II. Input file

```
<html>  
    <body>
```

```
<h1>Title</h1>
  this is practical 2 b
</body>
</html>
```

III. Output

```
<html>

<body>

<h1>
</h1>

</body>

</html>
```

IV. C. WRITE A LEX PROGRAM WHICH ADDS LINE NUMBERS TO THE GIVEN FILE AND DISPLAY THE SAME ONTO THE STANDARD OUTPUT.

I. Lex file

```
/* Program to add line numbers
to a given file*/
%{
  int line_number = 1; // initializing line number to 1
%}

/* simple name definitions to simplify
the scanner specification name
definition of line*/
line .*\\n

%%
{line} { printf("%10d %s", line_number++, yytext); }

%%
```

```
int yywrap(){  
  
int main(int argc, char*argv[])  
{  
    // yyin as pointer of File type  
    extern FILE *yyin;  
  
    /* yyin points to the file in.c and opens it in read mode.*/  
    yyin = fopen("in.c", "r");  
  
    yylex(); // The function that starts the analysis.  
  
    return 0;  
}
```

II. Input file

```
#include<stdio.h>  
  
#define PI 3.1415  
  
main() {  
    int i, j;  
    i = 5;  
    for (j=0; j<i*i; j++){  
        j = i+j;  
    }  
    int jarvis = PI * i * j;  
    printf("%d", jarvis);  
}
```

III. Output

```
1 #include<stdio.h>  
2  
3 #define PI 3.1415  
4  
5 main() {  
6     int i, j;  
7     i = 5;
```

```
8     for (j=0; j<i*i; j++){
9         j = i+j;
10    }
11    int jarvis = PI * i * j;
12    printf("%d", jarvis);
13 }
```

V. D. WRITE A LEX PROGRAM TO COUNT THE NUMBER OF COMMENT LINES IN A GIVEN C PROGRAM. ALSO ELIMINATE THEM AND COPY THAT PROGRAM INTO SEPARATE FILE.

I. Lex file

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    int a=0, b=0, c=0, d;
%}
%x COMMENT

%%
"//" .*    {a++;}
"/*"      { BEGIN COMMENT; }
<COMMENT>"*/" {c++; BEGIN INITIAL; }
<COMMENT>.    ;
%%

int yywrap(){
    return 1;
}

void main(int argc, char *argv[]){
    yyin=fopen("in.c", "r");
    yylex();
    printf("single line %d \nmultiline %d \n", a, c);
    d=a+c;
    printf("total %d \n", d);
}
```

II. Input file

```
#include<stdio.h>
// define pi for fun
```

```
#define PI 3.1415

main() {
    int i, j;
    // initialize i
    i = 5;
    for (j=0; j<i*i; j++){
        j = i+j;
    }
    /*
    this is some brainiac activity!!!!
    */
    int jarvis = PI * i * j; // this is algorithm to sink blackhole
    printf("%d", jarvis);
}
```

III. Output

```
#include<stdio.h>

#define PI 3.1415

main() {
    int i, j;

    i = 5;
    for (j=0; j<i*i; j++){
        j = i+j;
    }

    int jarvis = PI * i * j;
    printf("%d", jarvis);
}
single line 3
multiline 1
total 4
```