



LAB DAY 03

1. The lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Write a LEX specification file to take input C program from a .c file and count the number of characters, number of lines & number of words.

Input Source Program: (sample.c)

```
#include <stdio.h>
int main()
{
    int number1, number2, sum;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);
    sum = number1 + number2;
    printf("%d + %d = %d", number1, number2, sum);
    return 0;
}
```

Ans

```
#include <stdio.h>
int main()
{
    char str[100]; //input string with size 100

    int words=0, newline=0, characters=0; // counter variables

    scanf("%[^\n]", &str); //scanf formatting

    for(int i=0; str[i]!='\0'; i++)
    {
        if(str[i] == ' ')
        {
            words++;
        }
        else if(str[i] == '\n')
        {
            newline++;
            words++; //since with every next line new words start. corner case 1
        }
        else if(str[i] != ' ' && str[i] != '\n'){
            characters++;
        }
    }
}
```



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME
CSA14 COMPILER DESIGN



```
    }  
  }  
  if(characters > 0)//Corner case 2,3.  
  {  
    words++;  
    newline++;  
  }  
  printf("Total number of words : %d\n",words);  
  printf("Total number of lines : %d\n",newline);  
  printf("Total number of characters : %d\n",characters);  
  return 0;  
}
```

2. Write a LEX program to print all the constants in the given C source program file.

Input Source Program: (sample.c)

```
#define PI 3.14  
#include<stdio.h> #include<conio.h>  
void main()  
{  
  int a,b,c = 30;  
  printf("hello");  
}
```

Ans

```
digit [0-9]  
%{  
  int cons=0;  
  %}  
  %%  
  [0-9]+.[0-9]+ { cons++; printf("%s is a constant\n", yytext); }  
  .|\n { }  
  %%  
  int yywrap(void) {  
    return 1; }  
  int main(void)  
  {  
    FILE *f;  
    char file[10];  
    printf("Enter File Name : ");  
    scanf("%s",file);  
    f = fopen(file,"r");  
    yyin = f;  
    yylex();  
    printf("Number of Constants : %d\n", cons);
```



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME
CSA14 COMPILER DESIGN



```
fclose(yyin);  
}
```

3. Write a LEX program to count the number of Macros defined and header files included in the C program.

Input Source Program: (sample.c)

```
#define PI 3.14  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int a,b,c = 30;  
printf("hello");  
}
```

Ans

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

```
int macro_count = 0;  
int header_count = 0;  
%}
```

```
%%
```

```
"#define" {  
    macro_count++;  
    // Skip the rest of the line after #define  
    while (input() != '\n');  
    unput('\n');  
}
```

```
"#include" {  
    header_count++;  
    // Skip the rest of the line after #include  
    while (input() != '\n');  
    unput('\n');  
}
```

```
.\n {  
    // Match any other character and new lines to keep scanning the input  
}  
%%
```

```
int main(int argc, char **argv)  
{  
    if (argc > 1) {
```



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME
CSA14 COMPILER DESIGN



```
FILE *file = fopen(argv[1], "r");
if (!file) {
    perror(argv[1]);
    return 1;
}
yyin = file;
}

yylex();

printf("Number of macros defined: %d\n", macro_count);
printf("Number of header files included: %d\n", header_count);

return 0;
}
```

4. Write a LEX program to print all HTML tags in the input file.

Input Source Program: (sample.html)

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Ans

```
%{
int tags;
}%
%%
"<"[^>]*> { tags++; printf("%s \n", yytext); }
.|\\n { }
%%
int yywrap(void) {
return 1; }
int main(void)
{
FILE *f;
char file[10];
printf("Enter File Name : ");
scanf("%s", file);
f = fopen(file, "r");
yyin = f;
yylex();
}
```



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME
CSA14 COMPILER DESIGN



```
printf("\n Number of html tags: %d",tags);  
fclose(yyin);  
}
```

5. Write a LEX program which adds line numbers to the given C program file and display the same in the standard output.

Input Source Program: (sample.c)

```
#define PI 3.14  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int a,b,c = 30;  
printf("hello");  
}
```

Ans

```
%{  
#include <stdio.h>  
int line_num = 1;  
%}  
  
%%  
\n  {  
    // Print the line number at the beginning of each new line  
    printf("%d: %s", line_num, yytext);  
    line_num++;  
  }  
.|\\t {  
    // Print each character as it is, preserving the input format  
    printf("%s", yytext);  
  }  
%%
```

```
int main(int argc, char **argv)  
{  
    if (argc > 1) {  
        FILE *file = fopen(argv[1], "r");  
        if (!file) {  
            perror(argv[1]);  
            return 1;  
        }  
        yyin = file;  
    }  
}
```



```
yylex();  
return 0;  
}
```

6. Write a LEX program to count the number of comment lines in a given C program and eliminate them and write into another file.

Input Source File: (input.c)

```
#include<stdio.h>  
int main()  
{  
int a,b,c; /*variable declaration*/  
printf("enter two numbers");  
scanf("%d %d",&a,&b);  
c=a+b;//adding two numbers  
printf("sum is %d",c);  
return 0;  
}
```

Ans

```
%{  
int c=0;  
%}  
%%  
"/"[a-z]+ {c++;}  
"//"[a-z]+ {c++;}  
  
.+ {printf("The Given input is not comment line");}  
%%  
int yywrap(){}  
int main()  
{  
printf("enter : ");  
yylex();  
printf("\nnno of comments:%d",c);  
return 0;  
}
```

7. Implement Lexical Analyzer using LEX or FLEX (Fast Lexical Analyzer). The program should separate the tokens in the given C program and display with appropriate caption.

Input Source Program: (sample.c)

```
#include<stdio.h>  
void main()  
{  
int a,b,c = 30;  
printf("hello");  
}
```



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME
CSA14 COMPILER DESIGN



Ans

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

%%

if|else|while|int|switch|for|char { printf("\n%s is a KEYWORD", yytext);}
[a-zA-Z0-9]+ { printf("\n%s is IDENTIFIER", yytext);}

%%
int yywrap( ){}
int main()
{
printf("Enter the input:");
while(yylex());
}
```

8. Write a LEX specification counts the number of characters, number of lines & number of words.

Ans

```
%{
#include<stdio.h>
int lc=0,sc=0,ch=0,wc=0;
%}

%%

[\n] { lc++; ch+=yyleng;}
[ \t] { sc++; ch+=yyleng;}
[^\\t\\n ]+ { wc++; ch+=yyleng;}
%%

int yywrap()
{ return 1; }

int main(){
printf("Enter the Sentence : ");
yylex();
printf("Number of lines,words,char are :%d %d %d\\n",lc,wc,ch);
return 0;
}
```