**Lex Programs**

1. To count no. of vowels and consonants in the given input

%{

int vo,con;

%}

%%

[aeiouAEIOU] {vo++;}

[^aeiouAEIOU] {con++;}

%%

main()

{

yylex();

printf("Vowel count:%d, Consonent count:%d",vo,con);

}

1. To count no of + and - operators in the given input expression

%{

int pc,nc;

%}

%%

[+] {pc++;}

[-] {nc++;}

%%

main()

{

yylex();

printf("Positive count:%d, Negative count:%d",pc,nc);

}

1. To count number of characters, words, spaces and lines in the given file

%{

int ch=0,bl=0,li=0,wr=0;

%}

%%

[\n] {li++;wr++;}

[\t] {bl++;wr++;}

[" "] {bl++;wr++;}

[^\n\t] {ch++;}

%%

int main()

{

FILE \*fp;

char file[10];

printf("Enter file name: ");

scanf("%s",file);

fp=fopen(file,"r");

yyin=fp;

yylex();

printf("cc=%d\nbc=%d\nlc=%d\nwc=%d\n",ch,bl,li,wr);

return 0;

}

1. To separate comment lines in the given program

%{

int com=0;

%}

%%

"/\*"[^\n]+"\*/" {com++;fprintf(yyout," ");}

%%

int main()

{

printf("Write a C program\n");

yyout=fopen("output","w");

yylex();

printf("Comment=%d\n",com);

return 0;

}

1. a) to check for the validity of expression

%{

#include<stdio.h>

int a=0,s=0,m=0,d=0,ob=0,cb=0;

int flaga=0,flags=0,flagm=0,flagd=0;

%}

id [a-zA-Z]+

%%

[id] {printf("%s is an identifier\n",yytext);}

[+] {a++;flaga=1;}

[-] {s++;flags=1;}

[\*] {m++;flagm=1;}

[/] {d++;flagd=1;}

[(] {ob++;}

[)] {cb++;}

%%

int main()

{

printf("Enter exp\n");

yylex();

if(ob-cb==0)

{

printf("Valid expression\n");

}

else

{

printf("Invalid expression\n");

}

printf("Add=%d\nSub=%d\nMul=%d\nDiv=%d\n",a,s,m,d);

printf("Operators are: \n");

if(flaga)

{

printf("+\n");

}

if(flags)

{

printf("-\n");

}

if(flagm)

{

printf("\*\n");

}

if(flagd)

{

printf("/\n");

}

return 0;

}

b) to check for the validity of expression

%{

#include<stdio.h>

int a=0,s=0,m=0,d=0,ob=0,cb=0;

int flaga=0,flags=0,flagm=0,flagd=0;

%}

id [a-zA-Z]+

%%

[id] {printf("%s is an identifier\n",yytext);}

[+] {a++;flaga=1;}

[-] {s++;flags=1;}

[\*] {m++;flagm=1;}

[/] {d++;flagd=1;}

[(] {ob++;}

[)] {cb++;}

%%

int main()

{

printf("Enter exp\n");

yylex();

if(ob-cb==0)

{

printf("Valid expression\n");

}

else

{

printf("Invalid expression\n");

}

printf("Add=%d\nSub=%d\nMul=%d\nDiv=%d\n",a,s,m,d);

printf("Operators are: \n");

if(flaga)

{

printf("+\n");

}

if(flags)

{

printf("-\n");

}

if(flagm)

{

printf("\*\n");

}

if(flagd)

{

printf("/\n");

}

return 0;

}

1. To check whether the given sentence is simple or complex

%{

int flag=0;

%}

%%

(" "[aA][nN][dD]" ")|(" "[oO][rR]" ")|(" "[bB][uU][tT]" ") {flag=1;}

%%

int main()

{

printf("Enter a sentence\n");

yylex();

if(flag==1)

{

printf("Compound statement\n");

}

else

{

printf("Simple statement\n");

}

return 0;

}

1. To check valid identifiers and keywords in the given file and count the same

%{

#include<stdio.h>

int count=0;

%}

op [+-\*/]

letter [a-zA-Z]

digit [0-9]

id {letter}\*|({letter}{digit})+

notid ({digit}{letter})+

%%

[\t\n]+

("int")|("float")|("char")|("case")|("default")|("if")|("for")|("printf")|("scanf") {printf("%s is a keyword\n",yytext);}

{id} {printf("%s is an identifier\n",yytext);count++;}

{notid} {printf("%s is not an identifier\n",yytext);}

%%

int main()

{

FILE \*fp;

char file[10];

printf("Enter file name\n");

scanf("%s",file);

fp=fopen(file,"r");

yyin=fp;

yylex();

printf("Total identifiers: %d\n",count);

return 0;

}

1. [Lex program to check if characters other than alphabets occur in a string](https://www.geeksforgeeks.org/lex-program-to-check-if-characters-other-than-alphabets-occur-in-a-string/)

%{

  int len=0;

%}

// Rules to identify if a character apart from alphabets

// occurs in a string

%%

[a-zA-Z]+ {printf("No character other than alphabets");}

/\* here . will match any other character than alphabets

 because alphabets are already matched above

 \* will matches 0 or more characters in front of it.

\*/

.\* {printf("character other than alphabets present"); }

%%

// code section

int yywrap() { }

int main()

 {

  yylex();

  return 0;

 }

1. [Lex program to copy the content of one file to another file](https://www.geeksforgeeks.org/lex-program-to-copy-the-content-of-one-file-to-another-file/)

/\* LEX code to replace a word with another

   taking input from file \*/

/\* Definition section \*/

/\* character array line can be

   accessed inside rule section and main() \*/

%{

#include<stdio.h>

#include<string.h>

char line[100];

%}

/\* Rule Section \*/

/\* Rule 1 writes the string stored in line

   character array to file output.txt \*/

/\* Rule 2 copies the matched token

   i.e every character except newline character

    to line character array  \*/

%%

['\n']    { fprintf(yyout,"%s\n",line);}

(.\*)      { strcpy(line,yytext); line[0] = '\0'; }

<<EOF>> { fprintf(yyout,"%s",line); return 0;}

%%

int yywrap()

{

    return 1;

}

/\* code section \*/

int main()

{

        extern FILE \*yyin, \*yyout;

        /\* open the source file

           in read mode \*/

    yyin=fopen("input.txt","r");

        /\* open the output file

           in write mode \*/

    yyout=fopen("output.txt","w");

    yylex();

}

1. Lex Program to check valid Mobile Number

%{

    /\* Definition section \*/

%}

/\* Rule Section \*/

%%

[1-9][0-9]{9} {printf("\nMobile Number Valid\n");}

.+ {printf("\nMobile Number Invalid\n");}

%%

// driver code

int main()

{

    printf("\nEnter Mobile Number : ");

    yylex();

    printf("\n");

    return 0;

}

1. To count number of comments in the given input.

%

{

int com=0;

%}

%%

"/\*"[^\n]+"\*/" {com++;fprintf(yyout," ");}

%%

int main()

{

printf("Write a C program\n");

yyout=fopen("output","w");

yylex();

printf("Comment=%d\n",com);

return 0;

}

1. To check and count number of identifiers in the given program

%{

#include<stdio.h>

int count=0;

%}

op [+-\*/]

letter [a-zA-Z]

digit [0-9]

id {letter}\*|({letter}{digit})+

notid ({digit}{letter})+

%%

[\t\n]+

("int")|("float")|("char")|("case")|("default")|("if")|("for")|("printf")|("scanf") {printf("%s is a keyword\n",yytext);}

{id} {printf("%s is an identifier\n",yytext);count++;}

{notid} {printf("%s is not an identifier\n",yytext);}

%%

int main()

{

FILE \*fp;

char file[10];

printf("Enter file name\n");

scanf("%s",file);

fp=fopen(file,"r");

yyin=fp;

yylex();

printf("Total identifiers: %d\n",count);

return 0;

}