**1.LEX program to calculate number of vowels and consonants in a text**

%{

int ccount=0,vcount=0;

%}

%%

[aeiouAEIOU] vcount++;

[a-zA-z] ccount++;

%%

int main(){

printf("\nEnter the text to count vowels and consonants (Enter CTRL+D to stop): ");

yylex();

printf("\nNumber of Vowels = %d and Number of consonants = %d\n",vcount,ccount);

return 0;

}

**2.LEX program to calculate number of positive integer, negative integer, positive fraction and negative fraction**

%{

int nint=0,pint=0,nfrac=0,pfrac=0;

%}

%%

[+]?[0-9]+ pint++;

[-][0-9]+ nint++;

[+]?[0-9]\*[.][0-9]+ pfrac++;

[-][0-9]\*[.][0-9]+ nfrac++;

%%

int main(){

printf("\nEnter some numbers (Enter CTRL+D to stop) :");

yylex();

printf("\nNumber of positive integers = %d, number of negative integers = %d",pint,nint);

printf("\nNumber of positive fractions = %d, number of negative fractions = %d\n",pfrac,nfrac);

return 0;

}

**3.LEX program to count number of printf and scanf statement in a given C program.**

%{

int prfcount=0,scfcount=0,temp;

%}

%%

"printf" prfcount++;

"scanf" scfcount++;

%%

int main(int argc,char \*argv[]){

if(argc>1){

FILE \*fptr=fopen(argv[1],"r");

if(!fptr){

printf("\nCan't open the file!!\n");

return -1;

}

yyin=fptr;

yylex();

printf("\nNumber of printf = %d and number of scanf = %d\n",prfcount,scfcount);

}else{

printf("\nEnter a valid filename!!\n");

}

return 0;

}

**4.LEX program replace print by ‘read’ and scanf by ‘write’ and display the final outcome.**

%{

%}

%%

"printf" {fprintf(yyout,"%s","write");}

"scanf" {fprintf(yyout,"%s","read");}

%%

int main(int argc,char \*\*argv){

if(argc>1){

FILE \*in=fopen(argv[1],"r");

if(!in){

printf("Cannot open input file!!\n");

return 1;

}

yyin=in;

char filename[256]="changed\_";

strcat(filename,argv[1]);

FILE \*out=fopen(filename,"w");

if(!out){

printf("Cannot open output file!!\n");

return 1;

}

yyout=out;

yylex();

printf("Replced the statments\n");

}else printf("Enter a valid file name!!\n");

return 0;

}

**5.LEX program to count number of characters, words, spaces, lines in a file.**

%{

int charcount=0,wordcount=0,spaceccount=0,linecount=0;

%}

%%

[a-zA-Z0-9]+ {wordcount++;charcount+=yyleng;}

[\n] linecount++;

[\t ]+ spaceccount++;

. charcount++;

%%

int main(){

printf("\nEnter the text (Enter CTRL+D to stop): ");

yylex();

printf("\nNumber of characeters = %d and number of words = %d",charcount,wordcount);

printf("\nNumber of lines = %d and number of spaces = %d\n",linecount,spaceccount);

return 0;

}

**6.LEX program to count numbers (integer and fraction) greater than 42 from a list of numbers.**

%{

int count=0;

%}

%%

[+]?(42[.][0]\*[1-9][0-9]\*|4[3-9]([.][0-9]+)?|[5-9][0-9]([.][0-9]+)?|[1-9][0-9][0-9]+([.][0-9]+)?) count++;

%%

int main(){

printf("Enter the numbers (Enter CTRL+D to stop): ");

yylex();

printf("\nNumber of integers and fractions grater than 42 = %d\n",count);

return 0;

}

**7.LEX program to count comment lines and remove comment lines from a given C program.**

%{

int count=0;

%}

%x C

%%

"//".\*[\n] count++;

"/\*" {BEGIN C;}

<C>"\*/" {BEGIN 0; count++;}

<C>\n ;

<C>. ;

%%

int main(int argc,char \*\*argv){

if(argc>1){

FILE \*in=fopen(argv[1],"r");

if(!in){

printf("Cannot open input file!!\n");

return 1;

}

yyin=in;

char filename[256]="without\_comments\_";

strcat(filename,argv[1]);

FILE \*out=fopen(filename,"w");

if(!out){

printf("Cannot open output file!!\n");

return 1;

}

yyout=out;

yylex();

printf("Number of comments = %d\n",count);

}else printf("Enter a valid filename!!\n");

}

**8.LEX program to find the following regular expressions**

**a)Any string beginning with abb and ending with ccd.**

**b)A string with one or more occurrence of ab.**

%{

int a=0,b=0;

%}

%%

"abb"[a-z]\*"ccd" a=1;

[a-z]\*ab[a-z]\* b=1;

%%

int main(){

printf("Enter a string (Press CTRL+D to stop): ");

yylex();

if(a) printf("\nThe string starts with abb and ends with ccd\n");

else if(b) printf("\nString contains one or more occurance of 'ab'\n");

else printf("\nString not accenpted\n");

return 0;

}

**9.LEX program to find number of identifiers and number of integers and number of string which are neither identifier nor integer.**

%{

int idcount=0,intcount=0,stcount=0;

%}

%%

[a-zA-Z]([a-zA-Z]|[0-9])\* idcount++;

[+-]?[0-9]+ intcount++;

[^\t\n ]+ stcount++;

%%

int main(){

printf("Enter text (Enter CTRL+D to stop): ");

yylex();

printf("\nNumber of identifiers = %d, number of integers = %d and number of strings = %d",idcount,intcount,stcount);

return 0;

}

**10.LEX program to check whether a statement is simple or compound.**

%{

int iscompound=0;

%}

%%

[\t\n ]+ ;

AND|and|OR|or|BUT|but iscompound=1;

[^\t\n ]+ ;

%%

int main(){

printf("Enter a statment (Enter CTRL+D to stop): ");

yylex();

if(iscompound) printf("\nEntered statement is a compound statemenet\n");

else printf("\nEntered statement is a simple statement\n");

return 0;

}