Sample LEX programs:

(consider indentation appropriately)

**Steps to follow to execute the program:**

gedit programname.l [ Type the program]

lex programname.l

cc lex.yy.c -ll

./a.out

**Steps using flex & gcc**

edit in notepad, save as all programs, with .l option

flex name.l

gcc lex.yy.c // this produces a.exe

a

**LEX PROGRAMS**

**1:Program to count the number of vowels and consonants in a given string.**

%{

#include<stdio.h>

int vowels=0;

int cons=0;

%}

%%

[aeiouAEIOU] {vowels++;}

[a-zA-Z] {cons++;}

. {;}

%%

int yywrap()

{

return 1;

}

Int main()

{

printf("Enter the string.. at end press Ctrl d\n");

yylex();

printf("No of vowels=%d\nNo of consonants=%d\n",vowels,cons);

}

**2:Program to count the number of characters, words, spaces and lines in a given**

**input file**

%{

#include<stdio.h>

int c=0,w=0,s=0,l=0;

%}

WORD [^ \t\n,\.:]+

EOL [\n]

BLANK [ ]

%%

{WORD} {w++; c=c+yyleng;}

{BLANK} {s++;}

{EOL} {l++;}

%%

int yywrap()

{ return 1;

}

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

{

if(argc!=2)

{

printf("Usage: ./a.out p2in.txt\n");

exit(0);

}

yyin=fopen(argv[1],"r");

yylex();

printf("\nNo of characters=%d\nNo of words=%d\nNo of spaces=%d\nNo of

lines=%d\n",c,w,s,l);

}

3. **Program to count the no of comment lines in a given C program. Also eliminate**

**them and copy that program into separate file**

%{

#include<stdio.h>

int com=0;

%}

%s COMMENT

%%

"//".\* {com++;}

"/\*" {BEGIN COMMENT ;}

<COMMENT>"\*/" {BEGIN 0;com++ ;}

<COMMENT>\n {com++;}

<COMMENT>. {;}

.|\n {fprintf(yyout,"%s",yytext);}

%%

int yywrap()

{ return 1;

}

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

{

if(argc!=3)

{

printf("Usage: ./a.out input.txt out.txt\n");

exit(0);

}

yyin=fopen(argv[1],"r");

yyout=fopen(argv[2],"w");

yylex();

printf("No. of comment lines=%d\n",com);

}

**Input:input.txt**

#include<stdio.h>

main()

{ int a,b;

printf("Enter the value of a and b\n");

/\* Input values for a and b\*/

// Input values for a and b

scanf("%d%d",&a,&b);/\*values entered are assigned to a and b

vsdcfsd

dsg

sdb \*/

printf("The numbers are %d and %d\n",a,b)

printf("The numbers are %d and %d\n",a,b)

printf("The numbers are %d and %d\n",a,b);

}

**Output:out.txt**

#include<stdio.h>

main()

{ int a,b;

printf("Enter the value of a and b\n");

scanf("%d%d",&a,&b);

printf("The numbers are %d and %d\n",a,b)

printf("The numbers are %d and %d\n",a,b)

printf("The numbers are %d and %d\n",a,b);

}

4. **Program to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately**

%{

#include<stdio.h>

#include<string.h>

int noprt=0, nopnd=0, valid=1, top=-1,l=0, j=0;

char opnd[10][10], oprt[10][10], a[100];

%}

%%

"(" { top++; a[top]='(' ; }

"{" { top++; a[top]='{' ; }

"[" { top++; a[top]='[' ; }

")" { if(a[top]!='(')

{

valid=0; return;

}

else

top--;

}

"}" { if(a[top]!='{')

{

valid=0; return;

}

else

top--;

}

"]" { if(a[top]!='[')

{

valid=0; return;

}

else

top--;

}

"+"|"-"|"\*"|"/" { noprt++;strcpy(oprt[l], yytext);l++; }

[0-9]+|[a-zA-Z][a-zA-Z0-9]\* {nopnd++; strcpy(opnd[j],yytext);j++;}

%%

int yywrap()

{ return 1;

}

main()

{

int k;

printf("Enter the expression.. at end press Ctrl d\n");

yylex();

if(valid==1 && (nopnd-noprt)==1 && top==-1)

{

printf("The expression is valid\n");

printf("The operators are\n");

for(k=0;k<l;k++)

printf("%s\n",oprt[k]);

printf("Operands are\n");

for(k=0;k<j;k++)

printf("%s\n",opnd[k]);

}

else

printf("The expression is invalid\n");

}

5. **Program to recognize and count the number of identifiers in a given input file**

%{ #include<stdio.h>

int id=0;

%}

%%

[a-zA-Z][a-zA-Z0-9\_]\* {id++;ECHO;printf("\n");}

.+ {;}

\n {;}

%%

int yywrap()

{

return 1;

}

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

{

if(argc!=2)

{

printf("Usage: ./a.out p8in.txt\n");

exit(0);

}

yyin=fopen(argv[1],"r");

printf("Valid identifiers are\n");

yylex();

printf("No.of identifiers=%d\n",id);

}