***1. Lex program to count number of words.***

%{

#include<stdio.h>

#include<string.h>

int i = 0;

%}

%%

([a-zA-Z0-9])\* {i++;}

"\n" {printf("%d\n", i); i = 0;}

%%

int yywrap(void){}

int main()

{

yylex();

return 0;

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex count\_words.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy,c

C:\Users\user\DesktopFlex\_Program> a.exe

Utkarsh Malviya

2

Compiler Design Program

3

***2. Lex program to count words that are less than 10 & greater than 5.***

%{

#include<stdio.h>

int len=0, counter=0;

%}

%%

[a-zA-Z]+ { len=strlen(yytext);

if(len<10 && len>5)

{counter++;} }

%%

int yywrap (void )

{ }

int main()

{

printf("Enter the string:");

yylex();

printf("\n %d", counter);

return 0;

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex count\_len.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

Enter the string: Hello this is Compiler Design Program

3

# *3. LEX program to count the number of vowels and consonants in a given string.*

# %{ int vow\_count=0;

# int const\_count =0;

# %}

# %%

# [aeiouAEIOU] {vow\_count++;}

# [a-zA-Z] {const\_count++;}

# %%

# int yywrap(){}

# int main()

# { printf("Enter the string of vowels and consonents:");

# yylex();

# printf("Number of vowels are: %d\n", vow\_count);

# printf("Number of consonants are: %d\n", const\_count);

# return 0;

# }

***Output:***

C:\Users\user\DesktopFlex\_Program> flex vow\_cons.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

Enter the string of vowels and consonents: CompilerDesign

Number of vowels are: 5

Number of consonants are: 9

# *4. LEX program to count the frequency of the given word in a file.*

# %{

# #include<stdio.h>

# #include<string.h>

# char word [] = "lab";

# int count = 0;

# %}

# %%

# [a-zA-Z]+ { if(strcmp(yytext, word)==0)

# count++; }

# . ;

# %%

# int yywrap()

# {

# return 1;

# }

# int main()

# {

# extern FILE \*yyin, \*yyout;

# yyin=fopen("input.txt", "r");

# yylex();

# printf("%d", count);

# }

# **\* Input File: lab assignment for compiler design lab \***

***Output:***

C:\Users\user\DesktopFlex\_Program> flex frequency.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

2

# *5. LEX program to Identify and Count Positive and Negative Numbers.*

# %{

# int positive\_no = 0, negative\_no = 0;

# %}

# %%

# ^[-][0-9]+ {negative\_no++;

# printf("negative number = %s\n", yytext);}

# [0-9]+ {positive\_no++;

# printf("positive number = %s\n", yytext);}

# %%

# int yywrap(){}

# int main()

# {

# yylex();

# printf ("number of positive numbers = %d,"

# "number of negative numbers = %d\n", positive\_no, negative\_no);

# return 0;

# }

***Output:***

C:\Users\user\DesktopFlex\_Program> flex pos\_neg.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

12

positive number = 12

-15

negative number = -15

23

Positive number = 23

-87

Negative number = -87

number of positive number = 2 ,number of negative numbers = 2

# *6. LEX program to check whether given number is armstrong number or not.*

# %

# {

# #include <math.h>

# #include <string.h>

# void check(char\*);

# %

# }

# % %

# [0 - 9] + check(yytext);

# % %

# int main()

# {

# extern FILE\* yyin;

# yyin = fopen("num", "r");

# yylex();

# return 0;

# }

# void check(char\* a)

# {

# int len = strlen(a), i, num = 0;

# for (i = 0; i < len; i++)

# num = num \* 10 + (a[i] - '0');

# int x = 0, y = 0, temp = num;

# while (num > 0) {

# y = pow((num % 10), len);

# x = x + y;

# num = num / 10;

# }

# if (x == temp)

# printf("%d is armstrong number \n", temp);

# else

# printf("%d is not armstrong number\n", temp);

# }

# *Output:*

C:\Users\user\DesktopFlex\_Program> flex armstrong.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

153

153 is armstrong number

250

250 is not armstrong number

***7****. LEX code to count total number of tokens.*

%{

int n = 0 ;

%}

%%

"while"|"if"|"else" {n++;printf("\t keywords : %s", yytext);}

"int"|"float" {n++;printf("\t keywords : %s", yytext);}

[a-zA-Z\_][a-zA-Z0-9\_]\* {n++;printf("\t identifier : %s", yytext);}

"<="|"=="|"="|"++"|"-"|"\*"|"+" {n++;printf("\t operator : %s", yytext);}

[(){}|, ;] {n++;printf("\t separator : %s", yytext);}

[0-9]\*"."[0-9]+ {n++;printf("\t float : %s", yytext);}

[0-9]+ {n++;printf("\t integer : %s", yytext);}

. ;

%%

int main()

{

yylex();

printf("\n total no. of token = %d\n", n);

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex tokens.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

Int a=4,b=8

Keywords : int identifiers : a operator : = inteager : 4 separator : ,

identifiers : b operator : = inteager : 8

***8****. LEX program to count the Positive number,Negative number and Fractions.*

%{

int postiveno=0;

int negtiveno=0;

int positivefractions=0;

int negativefractions=0;

%}

DIGIT [0-9]

%%

\+?{DIGIT}+ postiveno++;

-{DIGIT}+negtiveno++;

\+?{DIGIT}\*\.{DIGIT}+ positivefractions++;

-{DIGIT}\*\.{DIGIT}+ negativefractions++;

. ;

%%

int main()

{

yylex();

printf("\nNo. of Positive numbers: %d", postiveno);

printf("\nNo. of Negative numbers: %d", negtiveno);

printf("\nNo. of Positive numbers in fractions: %d", positivefractions);

printf("\nNo. of Negative numbers in fractions: %d\n", negativefractions);

return 0;

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex pos\_neg\_frac.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

1 3 7 -3 6.5 -9 5.9 6 5.3 -1

No. of Positive numbers: 4

No. of Negative numbers: 3

No. of Positive numbers in fractions: 3

No. of Negative numbers in fractions: 0

***9****. LEX program to check whether a given number is even or odd.*

%{

#include<stdio.h>

int i;

%}

%%

[0-9]+ {i=atoi(yytext);

if(i%2==0)

printf("Even");

else

printf("Odd");}

%%

int yywrap(){}

int main()

{ yylex();

return 0;

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex even\_odd.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

18

Even

21

Odd

***10****. LEX program to check whether a number is Prime or Not.*

%{

#include<stdio.h>

#include<stdlib.h>

int flag,c,j;

%}

%%

[0-9]+ {c=atoi(yytext);

if(c==2)

{

printf("\n Prime number");

}

else if(c==0 || c==1)

{

printf("\n Not a Prime number");

}

else

{

for(j=2;j<c;j++)

{

if(c%j==0)

flag=1;

}

if(flag==1)

printf("\n Not a prime number");

else if(flag==0)

printf("\n Prime number");

}

}

%%

int main()

{

yylex();

return 0;

}

***Output:***

C:\Users\user\DesktopFlex\_Program> flex prime\_no.l

C:\Users\user\DesktopFlex\_Program> gcc lex.yy.c

C:\Users\user\DesktopFlex\_Program> a.exe

3

Prime number

14

Not a prime number