1..

%{

#undef yywrap

#define yywrap() 1

%}

%%

[\n] {

printf("Hello World\n");

}

%%

int main()

{

yylex(); //calling the rules section

}

2.

%{

/\*To find whether given letter is a vowel or not\*/

#undef yywrap

#define yywrap() 1

void display(int);

%}

%%

[a|e|i|o|u|] {

int flag=1;

display(flag);

return 0;

}

.+ {

int flag=0;

display(flag);

return 0;

}

%%

void display(int flag)

{

if(flag==1)

printf("The given letter [%s] is a vowel",yytext);

else

printf("The given letter [%s] is NOT a vowel",yytext);

}

int main()

{

printf("Enter a letter to check if it is a vowel or not");

yylex();

}

3.

%{

int no\_of\_lines = 0;

int no\_of\_chars = 0;

%}

/\*\*\*rule 1 counts the number of lines,

rule 2 counts the number of characters

and rule 3 specifies when to stop

taking input\*\*\*/

%%

\n ++no\_of\_lines;

. ++no\_of\_chars;

end return 0;

%%

/\*\*\* User code section\*\*\*/

int yywrap(){}

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

{

yylex();

printf("number of lines = %d, number of chars = %d\n",

no\_of\_lines, no\_of\_chars );

return 0;

}

4.

/\*lex program to count number of words\*/

%{

#include<stdio.h>

#include<string.h>

int i = 0;

%}

/\* Rules Section\*/

%%

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

number of words\*/

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

%%

int yywrap(void){}

int main()

{

// The function that starts the analysis

yylex();

return 0;

}

5.

%{

#include<stdio.h>

%}

%%

const |

final |

break |

else |

while |

for |

if |

do |

switch { printf(" %s is a keyword\n",yytext);

return(0); }

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

return(0); }

.\* { printf(" %s is not a identifier\n",yytext);

return (0);

}

%%

int yywrap()

{

return 1;

}

int main(){

printf("Enter a cahracter/word. \n");

yylex();

}

6. /\*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 yywrap()

{

return 1;

}

int main()

{

yylex();

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

}

7.

%{

#include<stdio.h>

int flag=0;

%}

%%

(""[nN][oO][rR]"")|(""[aA][nN][dD]"")|(""[oO][rR]"")|(""[sS][oO]"")|(""[bB][uU][tT]"") { printf("The compound word used is %s",yytext); flag=1;}

%%

int yywrap()

{

return 1;

}

int main()

{

printf("Enter the sentence\n");

yylex();

if(flag==1)

printf("\nCompound sentence\n");

else

printf("\nSimple sentence\n");

return 0;

}