21.count num of char and words

```
%{
int nlines,nwords,nchars;
%}

%%
\n {
 nchars++;nlines++;
 }

[^ \n\t]+ {nwords++, nchars=nchars+yyleng;}
. {nchars++;}
%%
int yywrap(void)
{
 return 1;
}
int main(int argc, char*argv[])
{
 yyin=fopen(argv[1],"r");
 yylex();
 printf("Lines = %d\nChars=%d\nWords=%d",nlines,nchars,nwords);

 return 0;
}
```

22.constants

```
%{
int vow=0;
int con=0;
%}
%%
[aeiouAEIOU1234567890!@#$%^&*()_+}{:"<>?|`=\;'/.,] {vow++;}
[a-zA-Z] {printf("%s\t",yytext);con++;}
%%
int yywrap(){}
int main(int argc,char*argv[])
{
 yyin=fopen(argv[1],"r");
 yylex();
 printf("no of consosnants is :%d\n",con);
 fclose(yyin);
}
```

23.macro

```
%{
int nmacro,nheader;
%}
%%
^#define {nmacro++;}
^#include {nheader++;}
```

```
.|\n {}
%%
int yywrap(void){
return 1;
}
int main(int argc,char* argv[]){
yyin=fopen(argv[1],"r");
yylex();
printf("No of macros =%d\n",nmacro);
printf("No of header=%d\n",nheader);
fclose(yyin);
}
```

24.html

```
%{
int c=0;
%}
%%
"<"[^>]*> {printf("%s",yytext);}
. {}
%%
int yywrap(void){}
int main(char argc[],char *argv[]){
yyin=fopen(argv[1],"r");
yylex();
fclose(yyin);
}
```

25.add line number

```
%{
int line_number = 1;  // initializing line number to 1
%}
line .*\n
%%
{line} { printf("%10d %s", line_number++, yytext); }

%%

int yywrap(){}

int main(int argc, char* argv[])
{
yyin = fopen(argv[1],"r");
yylex();
return 0;
}
```

26.comment line eliminate

```
%{
```

```
#include<stdio.h>
%}

%%

\/\/.* ;
\/\*(.*\n)*.*\*\/ ;
%%

int main()
{
yyin=fopen("input.c","r");
yylex();
return 0;
}

int yywrap()
{
return 1;
}
------------------------------------------------------------
```
27.THE CAPITAL WORDS FROM THE GIVEN INPUT.
```
%{
#include<stdio.h>
%}

%%
[A-Z]+[ \t\n] {printf("%s is a capital letter", yytext);}
. ;
%%
int main()
{
printf("\n Enter the input string:");
yylex();
}

int yywrap()
{
return 1;
}
------------------------------------------------------------
```
28.CHECK THE EMAIL ADDRESS IS VALID OR NOT.
```
%{
#include<stdio.h>
#include<ctype.h>
%}

%%
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,} { printf("Valid email address\n"); }
.                                              { printf("Invalid email address\n"); }

%%

int main(void)
{
   yylex();
   return 0;
}
```

29.WRITE A LEX PROGRAM TO CONVERT THE SUBSTRING abc TO ABC FROM THE GIVEN INPUT STR ING.

```
%{
#include<stdio.h>
int i;
%}
%%
[a-z A-Z]* {
for (i=0;i<=yyleng;i++)
{
 if((yytext[i]=='a')&&(yytext[i+1]=='b')&&(yytext[i+2]=='c'))
{
yytext[i]='A';
yytext[i+1]='B';
yytext[i+2]='C';
}
}
printf("%s",yytext);
}
[\t]* return;
.* {ECHO;}
\n {printf("%s",yytext);}
%%
main()
{
 yylex();
}
int yywrap()
{
 return 1;
}
```

30.MOBILE NUMBER

```
%{
#include<stdio.h>
%}

%%
[0-9]{10} { printf("Valid mobile number\n"); }
.        { printf("Invalid mobile number\n"); }

%%

int main(void)
{
   yylex();
   return 0;
}
```

31.token

```
%{
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);
}
int yywarp()
{}
```

32.vowels and consonants

```
%{
 #include<stdio.h>
 int vow=0, con=0;
%}

%%
[ \t\n]+;
[aeiouAEIOU]+    {vow++;}
[^aeiouAEIOU]     {con++;}
%%

int main( )
{
 printf("Enter the string:\n");
 yylex();
 printf("Number of vowels=%d\n",vow);
 printf("Number of consonants=%d\n",con);
}

int yywrap( )
{
 return 1;
}
```

33.no of constants

```
%{
```

```
int vow=0;
int con=0;
%}
%%
[aeiouAEIOU1234567890!@#$%^&*()_+}{:"<>?|`=\;'/.,] {vow++;}
[a-zA-Z] {printf("%s\n",yytext);con++;}
%%
int yywrap(){}
int main(int argc,char*argv[])
{
 yyin=fopen(argv[1],"r");
 yylex();
 printf("no of consosnants is :%d\n",con);
 fclose(yyin);
}
```

----

34. keyword

```
%{
%}

%%
bool|int|float|include|char|for|if|while|do|else|printf|scanf|main {}
%%
int yywrap(){}
int main(int argc,char*argv[])
{
   yyin=fopen("vowels.c","r");
   yyout=fopen("out.c","w");
   yylex();
   return 0;
}
```

35.number of keyword

```
%{
#include<stdio.h>
%}
%%

if |
else |
printf {printf("%s is a keyword\n", yytext);}
[0-9]+ {printf("%s is a number\n", yytext);}
[a-zA-Z]+ {printf("%s is a word\n", yytext);}
.|\nn {ECHO;}
%%
int main(){
printf("\n Enter the string: ");
yylex();
}
int yywrap()
```

```
{}
---------------------------------------------------------------------------------------------
36.count postive and negative

%{
int c=0;
int d=0;
%}
%%
[0-9] {c++;}
[-][0-9] {d++;}
%%
int yywrap(void){}
int main( char argc[],char *argv[]){
yyin=fopen(argv[1],"r");
yylex();
printf("%d,%d",c,d);
fclose(yyin);
}
---------------------------------------------------------------------------------------------
37.url
%%
((http)|(ftp))s?:\/\/[a-zA-Z0-9]+(\.[a-z]{2,})+(\/[a-zA-Z0-9+=?]*)* {printf("\nURL Valid\n");}

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

%%

int main()
{
 printf("\nEnter URL : ");
 yylex();
 printf("\n");
}
int yywrap()
{
   return 1;
}

---------------------------------------------------------------------------------------------
38.date of birth dob
%{
#include<stdio.h>
int i=0,yr=0,valid=0;
%}
%%
([0-2][0-9]|[3][0-1])\/((0(1|3|5|7|8))|(10|12))\/([1-2][0-9][0-9][-0-9]) {valid=1;}

([0-2][0-9]|30)\/((0(4|6|9))|11)\/([1-2][0-9][0-9][0-9]) {valid=1;}

([0-1][0-9]|2[0-8])\/02\/([1-2][0-9][0-9][0-9]) {valid=1;}

29\/02\/([1-2][0-9][0-9][0-9]) { while(yytext[i]!='/')i++; i++;while(yytext[i]!='/')i++;i++;while(i<yyleng)yr=(10*yr)
+(yytext[i++]-'0'); if(yr%4==0||(yr%100==0&&yr%400!=0))valid=1;}
```

```
%%
int main()
{
yyin=fopen("vpn.txt","r");
yylex();
if(valid==1) printf("It is a valid date\n");
else printf("It is not a valid date\n");
}
int yywrap()
{
return 1;
}
```

39.input digit

```
/* Lex program to check whether input is digit or not. */
%{
#include<stdio.h>
#include<stdlib.h>
%}
/* Rule Section */
%%
^[0-9]* printf("digit");
^[^0-9]|[0-9]*[a-zA-Z] printf("not a digit");
. ;
%%
int main()
{
  // The function that starts the analysis
 yylex();
  return 0;
}
int yywarp()
{
return 1;
}
```

40.calculator

```
%{

#undef yywrap
#define yywrap() 1
int f1=0,f2=0;
char oper;
float op1=0,op2=0,ans=0;
void eval();

%}

DIGIT [0-9]
NUM {DIGIT}+(\.{DIGIT}+)?
OP [*/+-]

%%
```

```
{NUM} {
 if(f1==0)
 {
  op1=atof(yytext);
  f1=1;
 }

 else if(f2==-1)
 {
  op2=atof(yytext);
  f2=1;
 }

 if((f1==1) && (f2==1))
 {
  eval();
  f1=0;
  f2=0;

 }
}

{OP} {

 oper=(char) *yytext;
 f2=-1;
}

[\n] {

 if(f1==1 && f2==1)
 {
  eval;
  f1=0;
  f2=0;
 }
}

%%


int main()
{
 yylex();
}


void eval()
{
 switch(oper)
 {
  case '+':
   ans=op1+op2;
   break;
  case '-':
   ans=op1-op2;
   break;

  case '*':
   ans=op1*op2;
   break;

  case '/':
   if(op2==0)
   {
    printf("ERROR");
    return;
   }
   else
   {
    ans=op1/op2;
   }
   break;
  default:
   printf("operation not available");
   break;
 }
 printf("The answer is = %lf",ans);
}
```