Exp1(identify capital letter)

```
%{
%}
%%
[A-Z]+ {printf("%s\n", yytext);}
.|\n {}

%%
int yywrap(){}
int main()
{
  printf("Enter a letter");
  yylex();
}
```

Exp2(digit or not)

```
%{
#include<stdio.h>
%}
%%
[0-9]+|[0-9]*\.[0-9]+ { printf("\n%s is DIGIT", yytext);}
.+ { printf("\n%s is NOT A DIGIT",yytext);}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 3(valid mobile number)

```
%{
%}
%%
[6-9][0-9]{9} {printf("\n mobile number valid\n");}
.+ {printf("\n mobile number invalid\n");
}
%%
int yywrap(){}
int main()
{
printf("\n enter the mobile number\n");
yylex();
}
```

Exp 4(vowels and consonant count)

```
%{
   int vow_count=0;
   int const_count =0;
%}
%%
[aeiouAEIOU] {vow_count++;}
[a-zA-Z] {const_count++;}
%%
int yywrap(){}
int main()
{
   printf("Enter the string:");
   yylex();
   printf("Number of vowels are:  %d\n", vow_count);
   printf("Number of consonants are:  %d\n", const_count);
   return 0;
}
```

Exp 5(recognise keywords and identifiers)

```
%{
#include<stdio.h>
%}
%%
if|else|while|int|switch|for|char { printf("\n%s is a KEYWORD", yytext);}
[a-zA-Z0-9]+ { printf("\n%s is IDENTIFIER", yytext);}
%%
int yywrap( ){}
int main()
{
        yylex();
}
```

Exp 6(positive and negative count)

```
%{
int positive_no = 0, negative_no = 0;
%}
%%
[-][0-9]+ {negative_no++;}
[0-9]+ {positive_no++;}
%%
int yywrap(){}
int main()
{
printf("enter number:\n");
yylex();
printf ("number of positive numbers = %d\n",positive_no);
printf("number of negative numbers = %d\n",negative_no);
return 0;
}
```

Exp 7(word and number)

```
%{
#include<stdio.h>
%}
%%
[0-9]+ {printf("\n%s is number",yytext);}
[a-zA-Z][a-zA-Z0-9]* {printf("\n%s is word",yytext);}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 8(accept string starting with vowels)

```
%{
%}
%%
[aeiouAEIOU].[a-zA-Z0-9.]+ {printf("Accepted");}
[a-zA-Z0-9]+ {printf("Not Accepted");}
%%
int yywrap(){}
int main()
{
 printf("enter string=\n");
 yylex();
}
```

Exp-9(length of longest string)

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

```
  int max_len = 0;
%}
%%
[a-zA-Z]+ {
  if (yyleng > max_len) {
    max_len = yyleng;
  }
}
.|\n {}
%%
int yywrap(){}
int main() {
  yylex();
  printf("Length of longest word: %d\n", max_len);
  return 0;
}
```

Exp-10(valid url)

```
%{
#include<stdio.h>
%}
%%
[http://]+[www.]+[a-z]+".com" {printf("\n valid url\n");}
.+ {printf("\n invalid url\n");}
%%
int yywrap()
{}
int main()
{
printf("enter url:\n");
yylex();
}
```

Exp-11(valid dob)

```
%{
%}
%%
[0-9][0-9]\/[0-1][0-9]\/[1-2][0-9]{3} { printf("valid");}
.+ { printf("invalid");}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 12(identify number,word,relational operator)

```
%{
#include<stdio.h>
%}
%%
[0-9]+ {printf("\n%s is number",yytext);}
[a-zA-Z][a-zA-Z0-9]* {printf("\n%s is word",yytext);}
">"|"<"|"<="|">="|"=="|"!=" {printf("\n%s is relational operator",yytext);}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 13(replace word)

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

```
%%
"apple"    { printf("orange "); }
.          { printf("%c", yytext[0]); }
%%

int yywrap(){}
int main()
{
   yylex();
}
```

Exp 14(simple calculator)

```
%{
#include<stdio.h>
float op1=6,op2=7;
%}

%%
"+" {printf("sum =%f",op1+op2);}
"-" {printf("diff=%f",op1-op2);}
"*" {printf("mul=%f",op1*op2);}
"/" {printf("div=%f",op1/op2);}
%%

int yywrap(){}
int main()
{
printf("enter proper operator.");
yylex();
}
```

Exp 15(valid mail)

```
%{
%}
```

```
%%
[a-zA-z0-9]+"@"[a-zA-Z]+".com" {printf("\n valid mail\n");}
.+ {printf("\n invalid mail\n");}
%%
int yywrap(){}
int main()
{
printf("enter the mail");
yylex();
}
```

Exp 16(abc to ABC)

```
%{
#include <ctype.h>
%}
%%
[a-z] { printf("%c", toupper(yytext[0])); }
.|\n { printf("%s", yytext); }
%%
int yywrap(){}
int main()
{
   yylex();
   return 0;
}
```

Exp 17(number of characters, number of lines & number of words.)

```
%{
int i =0,l=0,c=0;
%}
%%
[\n] {l++;}
[ ] {i++;}
[a-zA-Z0-9] {c++;}
```

```
%%
int yywrap(){}
int main()
{
 printf("enter the string: ");
 yylex();
 printf("no of lines:%d\n",l);
 printf("no of words is:%d",i+1);
 printf("no of characters:%d",c);
}
```

Exp 18(print all the constants)

```
%{
%}
%%
[0-9]+"."[0-9]+ {printf("%s is a floating-point constant\n", yytext); }
[0-9]+ {printf("%s is an integer constant\n", yytext); }
.|\n {}
%%
int yywrap(){}
int main()
{
 printf("Enter the code:");
 yylex();
}
```

Exp 19(count the number of Macros defined and header)

```
%{
int macro, header;
%}
%%
"#define" {macro++;}
"#include" {header++;}
```

```
.|\n {}
%%
int yywrap(){}
int main()
{
printf("enter the string:\n");
yylex();
printf("Number of macros= %d \n Number of headers= %d\n",macro,header);
}
```

Exp 20(print html tag)

```
%{
%}
%%
\<[^>]*\> {printf("%s\n is html tag",yytext);}
.|\n;
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 22(count comment lines)

```
%{
#include<stdio.h>
int c=0;
%}
%%
"/"[a-zA-Z0-9]+"/"  {c++;}
"//"[a-zA-Z0-9]+"\n"  {c++;}
%%
```

```
int yywrap( ){}
int main()
{
printf("enter the comment lines:\n");
yylex( );
printf("The number of comment lines=%d\n",c);
}
```

Exp 23(separation of token)

```
%{
#include<stdio.h>
%}
%%
bool|int|float|main|printf {printf("\n%s is a Keyword",yytext);}
[-,+]?[0-9]+ {printf("\n%s is a Constants",yytext);}
[,.'"]+ {printf("\n%s is a Punctuation Chars",yytext);}
[!@#$%^&*()]+ {printf("\n%s is a Special Chars",yytext);}
[a-zA-Z]+ {printf("\n%s is a Identifiers",yytext);}
%%
int yywrap(){}
int main()
{
yylex();
}
```

Exp 25(frequency of string)

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

```
%%
[a-zA-Z]+ { printf("Length of %s: %d\n", yytext, (int)strlen(yytext)); }
.|\n;
%%
int yywrap(){}
int main()
{
   yylex();
}
```