Set1

1. Write a LEX program to accept string starting with vowel.

Code:

```
%{
#include <stdio.h>
%}
%%
[AEIOUaeiou][a-zA-Z]*   { printf("Valid string: %s\n", yytext); }
[a-zA-Z]+            { printf("Invalid string: %s\n", yytext); }
%%
int main() {
   printf("Enter a string: ");
   yylex();
   return 0;
}
int yywrap() {
   return 1;
}
```

2. School management wants to validate DOB of all students. Write a LEX program to implement it.

Code:

```
%{
#include <stdio.h>
#include <stdlib.h>
%}
%%
(0[1-9]|[12][0-9]|3[01])-(0[1-9]|1[0-2])-[0-9]{4}  { printf("Valid DOB: %s\n", yytext); }
[0-9-]+                        { printf("Invalid DOB: %s\n", yytext); }
```

```
%%

int main() {

   printf("Enter DOB (DD-MM-YYYY): ");

   yylex();

   return 0;

}

int yywrap() {

   return 1;

}
```

3. Write a LEX program to count the number of comment lines in a given C program and eliminate them and write into another file.

Input Source File: (input.c)

```
#include<stdio.h>

int main()

{

int a,b,c; /*varible declaration*/ printf(“enter two numbers”); scanf(“%d %d”,&a,&b); c=a+b;//adding two numbers printf(“sum is %d”,c);

return 0;

}
```

Code:

```
%{

int com=0;

%}

%s COMMENT

%%

"/*" {BEGIN COMMENT;}

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

<COMMENT>\n {com++;}
```

```
<COMMENT>. {;}
\/\/.* {; com++;}
.|\n {fprintf(yyout,"%s",yytext);}
%%
void main(int argc, char *argv[])
{
if(argc!=3)
{
printf("usage : a.exe input.c output.c\n");
exit(0);
}
yyin=fopen(argv[1],"r");
yyout=fopen(argv[2],"w");
yylex();
printf("\n number of comments are = %d\n",com);
}
int yywrap()
{
return 1;
}
```

3. Implement a C program to eliminate left recursion.

```
Code:
#include <stdio.h>
#include <string.h>
void eliminateLeftRecursion(char nonTerminal, char alpha[], char beta[])
{
   printf("Grammar after eliminating left recursion:\n");
   printf("%c -> %s%c'\n", nonTerminal, beta, nonTerminal);
   printf("%c' -> %s%c' | ε\n", nonTerminal, alpha, nonTerminal);
}
int main() {
```

```
    char nonTerminal;
    char alpha[10], beta[10];
    printf("Enter the non-terminal: ");
    scanf(" %c", &nonTerminal);
    printf("Enter left-recursive production (A -> Aα | β), where α and β are
strings:\n");
    printf("Enter α: ");
    scanf("%s", alpha);
    printf("Enter β: ");
    scanf("%s", beta);
    eliminateLeftRecursion(nonTerminal, alpha, beta);
    return 0;
}
```