1a.
%x COMMENT
as well as %s COMMENT give same output in given input.c

```
           |  ^
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p1.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c -ll
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
No. of Single-line comments = 3
No. of Multi-line comments = 2
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ▮
```

 %s multiple states active at the same time
%x only one state active at a time

tried with

```
 2 #include <stdio.h>
 3 int main() {
 4 int a, b, c;
 5 /* This is a
 6     //single line comment inside multiline comment
 7 multi-line comment */
 8 a = b+ c;
 9 printf("%d", a);
10
11 return 0;
12 }
```

```
No. of Multi-line comments = 2
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ gedit p1.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ gedit input.c
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p1.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c -ll
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
No. of Single-line comments = 1
No. of Multi-line comments = 1
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ gedit p1.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p1.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c -ll
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
No. of Single-line comments = 0
No. of Multi-line comments = 1
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ▮
```

1-1 for %x and 0-1 for %s

1b
```
%{
#include <stdio.h>
int s_comment=0;
int m_comment=0;
%}
%x COMMENT
%%
"//"(.)* {s_comment++; }
"/*" { BEGIN(COMMENT); } /* Enter COMMENT state */
<COMMENT>. { }
<COMMENT>"*/" { m_comment++; BEGIN(0); } /* Exit COMMENT state */
<COMMENT>\n { /* Handle new lines inside comments */ }
.|\n {fprintf(yyout,"%s",yytext);}
%%
int main() {
yyin=fopen("input.c", "r");
yyout=fopen("output.c","w");
yylex();
printf("No. of Single-line comments = %d\n", s_comment);
printf("No. of Multi-line comments = %d\n", m_comment);
return 0;
}
```

2

```
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string :aaaabc
T_AB: aaaa
T_B: b
T_C: c
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string :aaaaad
aaaaaduser@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string :abcab
T_AB: a
T_B: b
T_C: c
T_AB: a
T_B: b
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string :aacaacbc
T_AC: aa
T_C: c
T_AC: aa
T_C: c
T_B: b
T_C: c
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

3

with reject

```
         ^~~~
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
pinkink
pinkinknumber of occurrences of PINK 1
number of occurrences of PIN 1
number of occurrences of INK 2
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
                                          \n  return;
```

without reject

```
         ^~~~
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
pinkink
number of occurrences of PINK 1
number of occurrences of PIN 0
number of occurrences of INK 1
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
                                          user@acl-ThinkCentre-M720t:~/Des
```

4

terminates only when all lower case string is encountered

```
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
bits ok Pil
Lower case detected:
bits
Begin yyterminate
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
bits hm
Lower case detected:
bits
Begin yyterminate
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Bits ok
Mixed case detected:
Bits Lower case detected:
ok
Begin yyterminate
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

```
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
BitsPilani
Mixed case detected:
BitsPilani
bitspilani
Lower case detected:
bitspilani
Begin yyterminate
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
BitsPilani
Mixed case detected:
BitsPilani
bitspilanIh
Mixed case detected:
bitspilanIh
hmmm
Lower case detected:
hmmm
Begin yyterminate
Lexing complete. Exiting...
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

5

first input from terminal second from file

```
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p6.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c -ll
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Starting first lex operation (taking input from terminal):
******************************
Enter a string : hmmmm ok Bits Pilani

Lower case token found= hmmmm
End of first yylex operation.
******************************
Switching input to 'input.txt'...

Starting second lex operation from 'input.txt':
Welcome To Compiler Construction
Lower case token found= course
End of second lex operation.
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

6
yyrestart(fp2);

7

yyin=fp2; doesn't change yyin pointer..



8

```
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string:hello Im bits Pilani
Read character by the regular expression: h
Read character by input(): e
Read character by the regular expression: l
Read character by input(): l
Read character by the regular expression: o
Read character by input():
Read character by the regular expression: I
Read character by input(): m
Read character by the regular expression:
Read character by input(): b
Read character by the regular expression: i
Read character by input(): t
Read character by the regular expression: s
Read character by input():
Read character by the regular expression: P
Read character by input(): i
Read character by the regular expression: l
Read character by input(): a
Read character by the regular expression: n
Read character by input(): i
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

9

```
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p8.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string: hello
Lower case token is = hello
The token after unput is = helun

The unput char is = un
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

10

```
The unput char is = un
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ gedit p9.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p9.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string: BITSPilani
First match: BITS
Combined: BITSPilani
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
Enter a string: BITSPilani
```

11

```
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ lex p10.l
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ cc lex.yy.c -ll
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string: 12345
Matched number: 12345
First 3 digits: 123
Rest of the number will be rescanned.
the last match: 45
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$ ./a.out
Enter a string: 1114567
Matched number: 1114567
First 3 digits: 111
Rest of the number will be rescanned.
Matched number: 4567
First 3 digits: 456
Rest of the number will be rescanned.
the last match: 7
user@acl-ThinkCentre-M720t:~/Desktop/2022a7ps1672h$
```

12

email id check

```
%{
#include<stdio.h>
%}
%%
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]+ { printf("valid email.. username: %s\n", yytext);
(.)* {printf("invalid\n");}
. ECHO;
\n return 1;
%%




int main()
{
printf("Enter a string :");
yylex();
return 0;
}
```