21-09-2021 COMPILER DESIGN

SACHIN RAGHUL T

OBSERVATION - 3

2019103573

1. write a lex program to identify whether a given symbol is operator symbol or not and identify its token name.

```
%option noyywrap
%{
    #include <stdio.h>
    int flag = 1;
%}

%%
[+-/*]{
   switch(yytext[0]){
       case '/' : printf("Division operator\n"); flag=0; break;
       case '+' : printf("Addition operator\n"); flag=0; break;
       case '-' : printf("Subtraction operator\n"); flag=0; break;
       case '*' : printf("Multiplication operator\n"); flag=0; break;
   }
}
[^+-/*'\n"]{
      printf("Not an operator\n");
}
%%
int main()
{
    yylex();
    return 0;
}
```

2. Write a Lex program to identify whether a given line is a comment or not.

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

%%
(/\*([^*]|[\r\n]|(\*+([^*/]|[\r\n])))*\*+/)|(//.*){
    printf("commented line\n");
}
%%
int main()
{
    yylex();
    return 0;
}
```

3. Write a lex program to recognize strings under 'a*', 'a*b+', 'abb'.

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

%%
a*{
   printf("%s matches the regEx : a*\n", yytext);
}
abb{
   printf("%s matches the regEx ; abb\n", yytext);
}
a*b+{
   printf("%s matches the regEx : a* b+ \n", yytext);
}
%%
int main()
{
   yyin = fopen("z3test.txt", "r");
   yylex();
   return 0;
}
```