```
/************************************************************

Name  : Shubham S. Tembhurkar
Sub   : CCL
PRN   : 1641060
Class : L.Y. Computer
Batch : B1
Aim   :Write a C program to simulate lexical analyzer for validating
operators.

************************************************************/

#include<stdio.h>
int main()
{
      char s[10];
      int c;
      do
      {
      printf("Enter any operator:");
      scanf("%s",s);
      switch(s[0])
      {
            case '<':if(s[1]=='=')
                        printf("\nless than or equal\n");
                  else
                        printf("\nless than");
                  break;
            case '>':if(s[1]=='=')
                        printf("\ngreater than or equal");
                  else
                        printf("\ngreater than");
                  break;
            case '+':if(s[1]=='+')
                        printf("\nunary increament operator");
                  else
                        printf("\nadd is an binary arithmatic operator");

                  break;
            case '-':if(s[1]=='-')
                        printf("\nunary decreament operator");
                  else
                        printf("\nminus is an binary arithmatic
operator");
                  break;
            case '/':if(s[1]=='*')
                        printf("\nit is not an operator");
                  else
                        printf("\ndivision is an binary arithmatic
operator");
                  break;
```

```
            case '*':printf("\nmultiplication is an binary arithmatic
operator");
                  break;

            case '%':printf("\nmodulus is an arithmatic operator");
                  break;
            case '!':if(s[1]=='=')
                        printf("\nnot equal");
                  else
                        printf("\nbit not");
                  break;
            case '=':if(s[1]=='=')
                        printf("\nit is an comparison operator");
                  else
                        printf("\nassignment operator");
                  break;
            case '&':if(s[1]=='&')
                        printf("\nlogical AND");
                  else
                        printf("\nBitwise AND");
                  break;
            case '|':if(s[1]=='|')
                        printf("\nlogical OR");
                  else
                        printf("\nBitwise OR");
                  break;
            case '~':printf("\nnegation operator");
                  break;
            case '?':if(s[1]==':')
                        printf("\nternary oprator is an unary oprator");
                  else
                        printf("\nnot an oprator");
                  break;
            default:printf("\nInvalid input!!");
                  break;
      }
      printf("\nDo you want to continue 1/0\n");
            scanf("%d",&c);
      }
      while(c==1);
      return(0);
}

/********************** OUTPUT ***************************

shubh@ubuntu:~/CCL$ gcc cc4.c

shubh@ubuntu:~/CCL$ ./a.out
Enter any operator:+

add is an binary arithmatic operator
Do you want to continue 1/0
```

```
1
Enter any operator:-

minus is an binary arithmatic operator
Do you want to continue 1/0
1
Enter any operator:>

greater than
Do you want to continue 1/0
1
Enter any operator:<

less than
Do you want to continue 1/0
1
Enter any operator:=

assignment operator
Do you want to continue 1/0
1
Enter any operator:/

division is an binary arithmatic operator
Do you want to continue 1/0
1
Enter any operator:*

multiplication is an binary arithmatic operator
Do you want to continue 1/0
1
Enter any operator:<=

less than or equal

Do you want to continue 1/0
1
Enter any operator:>=

greater than or equal
Do you want to continue 1/0
0
shubh@ubuntu:~/CCL$

**************************************************/
```