

PRACTICAL-4

AIM: Write a C program to stimulate lexical analyser for validating operator.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    int l,i,n,t,h;
    char a[30];
    printf("Enter any string:");
    gets(a);
    l = strlen(a);
    n=1;
    for(i=0;i<l;i++)
    {
        if(a[i]=='&')
        {
            if((a[i]=='&' && a[i+1]=='&')&& i+1<n)
            {
                printf("&& : Logical AND operator\n"); i++;
            }
            else
            {
                printf("%c : Bitwise AND operator\n",a[i]);
            }
        }
        if(a[i]=='|')
```



```
{
    if((a[i]=='|' && a[i+1]=='|')&& i+1<n)
    {
        printf("| : Logical OR operator\n"); i++;
    }
    else
    {
        printf("%c : Bitwise OR operator\n",a[i]);
    }
}

if(a[i]=='^')
{
    if((a[i]=='^' && a[i+1]=='=')&& i+1<n)
    {
        printf("^= : Assignment operator\n"); i++;
    }
    else
    {
        printf("%c : Bitwise EX-OR operator\n",a[i]);
    }
}

if(a[i]=='+')
{
    if((a[i]=='+' && a[i+1]=='+'))
    {
        printf("++ : Increment operator\n");i++;
    }
    else
```



```
        {
            printf("+ : Addition\n"); }
    }
    if(a[i]=='-')
    {
        if((a[i]=='-') && (a[i+1]=='-'))
        {
            printf("-- : Decrement operator\n"); i++;
        }
        else
        {
            printf("- : Subtraction\n");
        }
    }
    if(a[i]=='*')
    {
        printf("* : Multiplication\n");
    }
    if(a[i]=='/')
    {
        printf("/ : Division\n");
    }
    if(a[i]=='%')
    {
        printf("% : Modulo\n");
    }
    if(a[i]=='>')
    {
        if((a[i]=='>') && (a[i+1]=='='))
```



```
        {printf(">= : Greater Than or Equal to\n"); i++;}
else if((a[i]=='>')&&(a[i+1]=='>'))
    {
        printf(">> : Right Shift\n");i++;
    }
else
    {
        printf("> : Greater Than\n");
    }
}
if(a[i]=='<')
{
    if((a[i]=='<')&&(a[i+1]=='<')&&i+1<n)
    {
        printf("<= : Less Than or Equal to\n");i++;
    }
    else if((a[i]=='<')&&(a[i+1]=='<')&&i+1<n)
    {
        printf("<< : Left Shift\n");i++;
    }
    else
    {
        printf("< : Less Than\n");
    }
}
if(a[i]=='=')
{
    if((a[i]=='=' && a[i+1]=='=')
    {
```

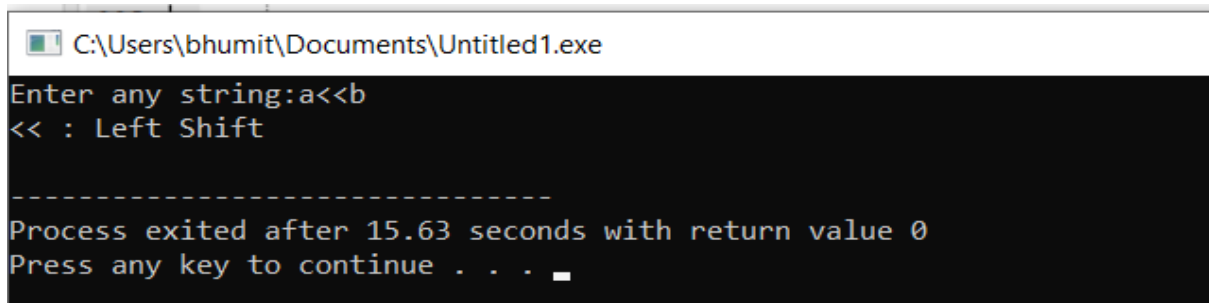


```
        printf("== : Equal To\n");i++;
    }
    else
    {
        printf("= : Assignment\n"); } } }

return 0;
}
```

OUTPUT:

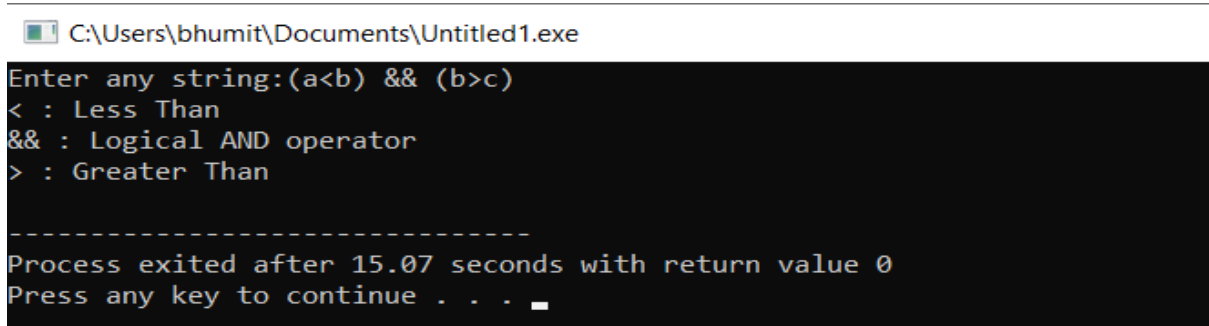
TEST CASE - 1:



```
C:\Users\bhumit\Documents\Untitled1.exe
Enter any string:a<<b
<< : Left Shift

-----
Process exited after 15.63 seconds with return value 0
Press any key to continue . . .
```

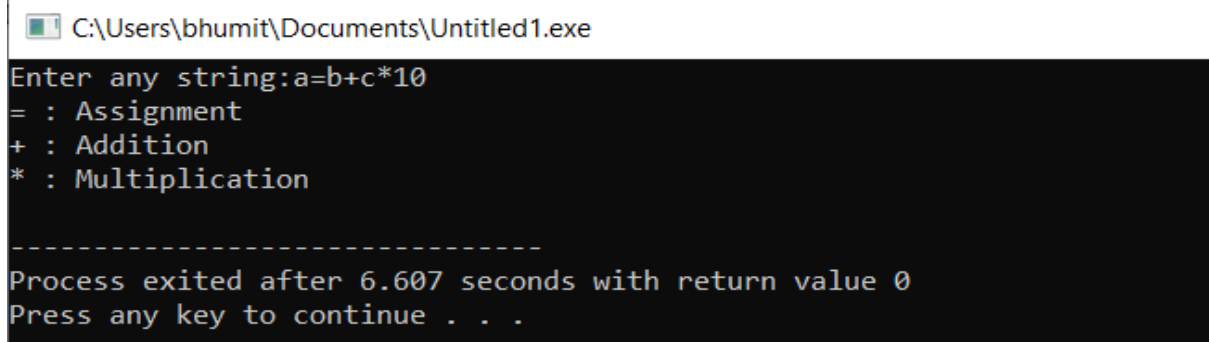
TEST CASE -2:



```
C:\Users\bhumit\Documents\Untitled1.exe
Enter any string:(a<b) && (b>c)
< : Less Than
&& : Logical AND operator
> : Greater Than

-----
Process exited after 15.07 seconds with return value 0
Press any key to continue . . .
```

TEST CASE -3:



```
C:\Users\bhumit\Documents\Untitled1.exe
Enter any string:a=b+c*10
= : Assignment
+ : Addition
* : Multiplication

-----
Process exited after 6.607 seconds with return value 0
Press any key to continue . . .
```

