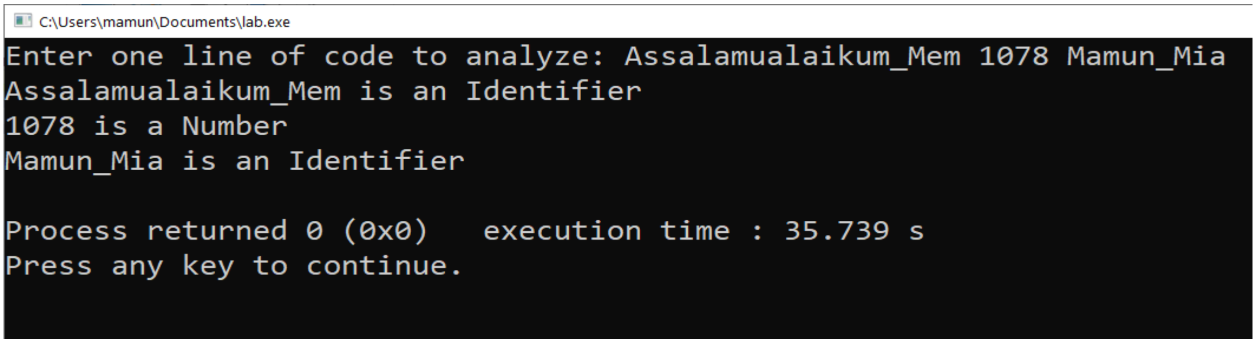
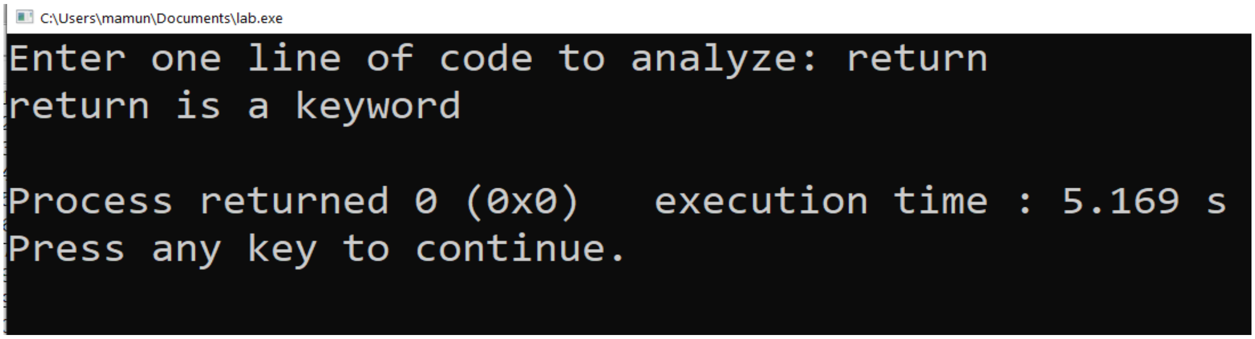
1. Question : Write a C program to implement a Lexical Analyzer.

```
#include<stdio.h>
#include<string.h>
void main(){
   char a[5];
   printf("Enter an Operator\n");
   gets(a);
   int len= strlen(a);
   switch(a[0]){
   case '>':
      if(len==1) printf("This is greater than");
      else if(a[1]=='=') printf("This is greater equal");
      else printf("This is Invalid operator");
      break;
   case '<':
      if(len==1) printf("This is less than");
      else if(a[1]=='=') printf(" This is less< equal");
      else printf("This is  invalid Operator");
      break;
   case '=':
      if(len==1) printf("This is assignment Operator");
      else if(a[1]=='=') printf("This is equal to");
      else printf("This is operator is invalid");
      break;
   case '!':
      if(len==1) printf("This is negation Operator");
      else if(a[1]=='=') printf("This is not equal");
      else printf("This is operator is invalid");
      break;
   case '&':
      if(len==1) printf("This is Bit wise AND");
      else if(a[1]=='&') printf("This is logical AND Operator");
      else printf("This is operator is invalid");
      break;
   case '|':
      if(len==1) printf("This is Bit wise OR");
      else if(a[1]=='|') printf(" This is logical OR");
      else printf("This is operator is invalid");
      Break;
  case '+':
        if(len==1) printf("\nAddition");
        else if(a[1]=='+') printf("This is Increment Operator");
        else printf("This is operator is invalid");
        break;
    case '-':
        if(len==1) printf("This is Subtraction");
        else if(a[1]=='-') printf("This is Decrement Operator");
        else printf("This is  operator is invalid");
        break;
    case '*': printf(" This is  Multipication operator");
        break;
    case '/':
        printf("This is  Division operator");
        break;
    case '%':
        printf(" This is Modulas operator");
        break;
    default:
        printf("This is  Invalid Operator");
    }
    return 0;
}
```

Answer is :-



```
Enter one line of code to analyze: Assalamualaikum_Mem 1078 Mamun_Mia
Assalamualaikum_Mem is an Identifier
1078 is a Number
Mamun_Mia is an Identifier

Process returned 0 (0x0)   execution time : 35.739 s
Press any key to continue.
```

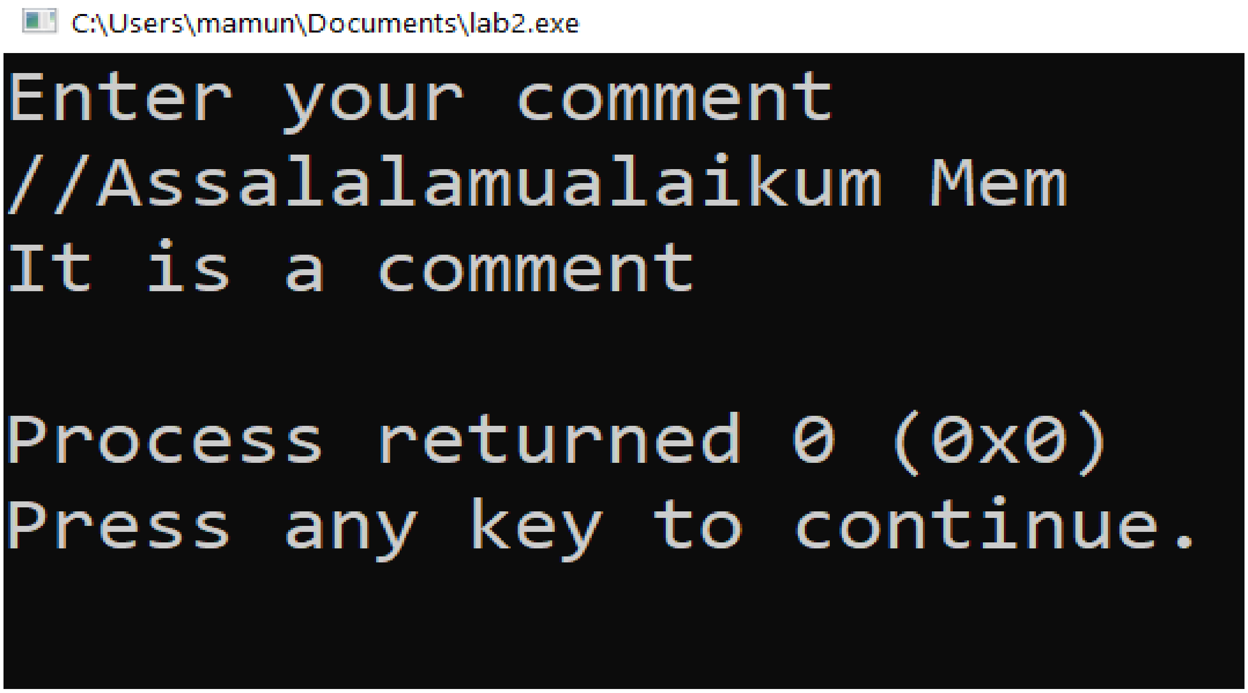


```
Enter one line of code to analyze: return
return is a keyword

Process returned 0 (0x0)   execution time : 5.169 s
Press any key to continue.
```

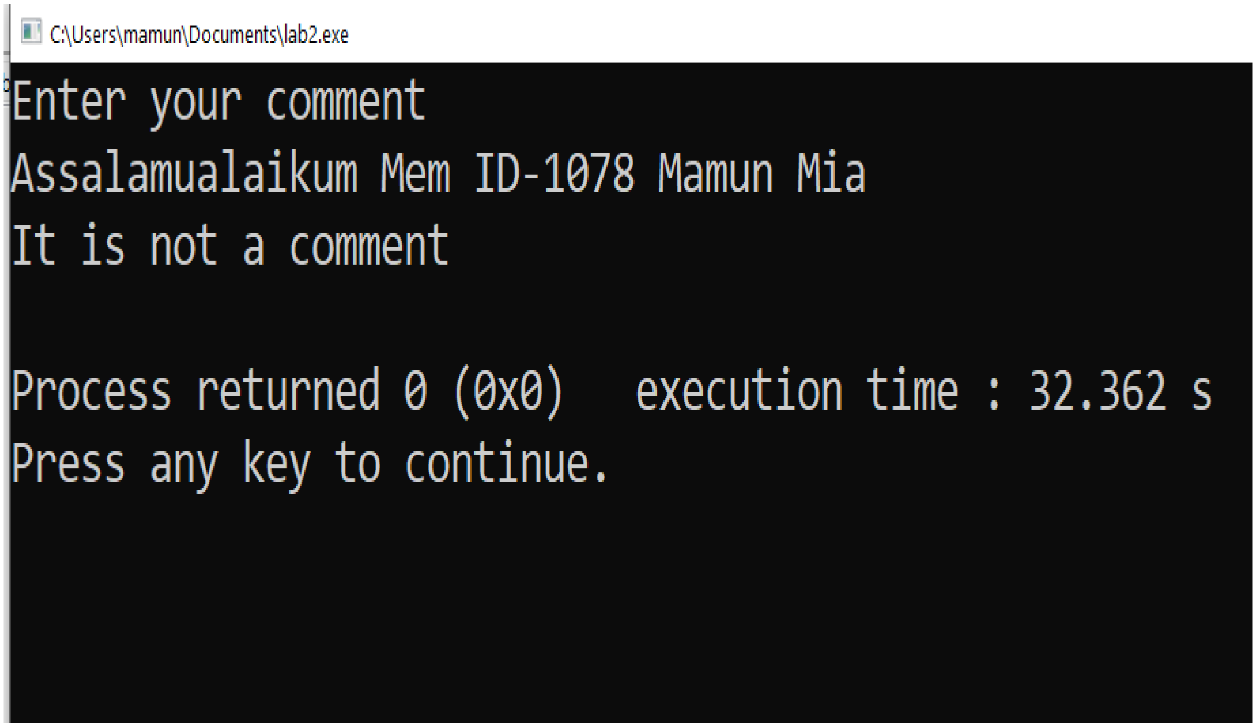
2. Question : Write a C program to identify whether a given line is a comment or not.

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char comment[100];
    int m, a=0;// m is intialization.
    printf("Enter your comment\n");
    gets(comment);
    if (comment[0]=='/'){
        if(comment[1]=='/'){
            printf("It is a comment\n");
        }
        else if (comment[1]=='*'){
            for(m=2; m<=100; m++){
                if(comment[m]=='*' && comment[m+1]=='/'){
                    printf("It is a comment\n");
                    a=1;
                    break;
                }else
                    continue;
            }
            if(a==0){
                printf("It is not a comment\n");
            }
        }else
        printf("It is not a comment\n");
    }else
        printf("It is not a comment\n");
}
```

Answer is :-



```
Enter your comment
//Assalalamualaikum Mem
It is a comment

Process returned 0 (0x0)
Press any key to continue.
```



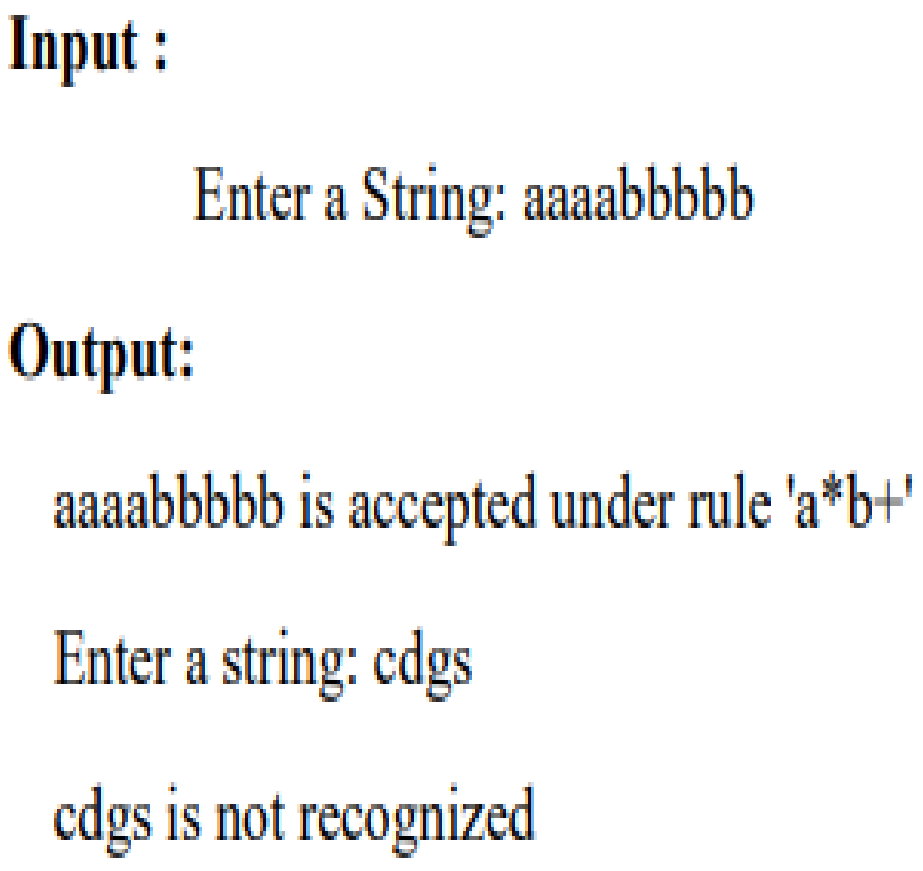
```
Enter your comment
Assalamualaikum Mem ID-1078 Mamun Mia
It is not a comment

Process returned 0 (0x0)   execution time : 32.362 s
Press any key to continue.
```

3. Question : Write a C program to recognize strings under ‘a’,’a*b+’,’abb’.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
char s[20],c;
int state=0,i=0;
clrscr();
printf("\n Enter a string:");
gets(s);
while(s[i]!='\0')
{
switch(state)
{
case 0: c=s[i++];
if(c=='a')
state=1;
else if(c=='b')
state=2;
else
state=6;
break;
case 1: c=s[i++];
if(c=='a')
state=3;
else if(c=='b')
state=4;
else
state=6;
break;
case 2: c=s[i++];
if(c=='a')
state=6;
else if(c=='b')
state=2;
else
state=6;
break;
case 3: c=s[i++];
if(c=='a')
state=3;
else if(c=='b')
state=2;
else
state=6;
break;
case 4: c=s[i++];
if(c=='a')
state=6;
else if(c=='b')
state=5;
else
state=6;
break;
case 5: c=s[i++];
if(c=='a')
state=6;
else if(c=='b')
state=2;
else
state=6;
break;
case 6: printf("\n %s is not recognised.",s);
exit(0);
}
```

```
}
 if(state==1)
printf("\n %s is accepted under rule 'a'",s);
else if((state==2)||(state==4))
printf("\n %s is accepted under rule 'a*b+'",s);
else if(state==5)
printf("\n %s is accepted under rule 'abb'",s);
getch();
}
```

Answer is :



Input :

```
Enter a String: aaaabbbbb
```

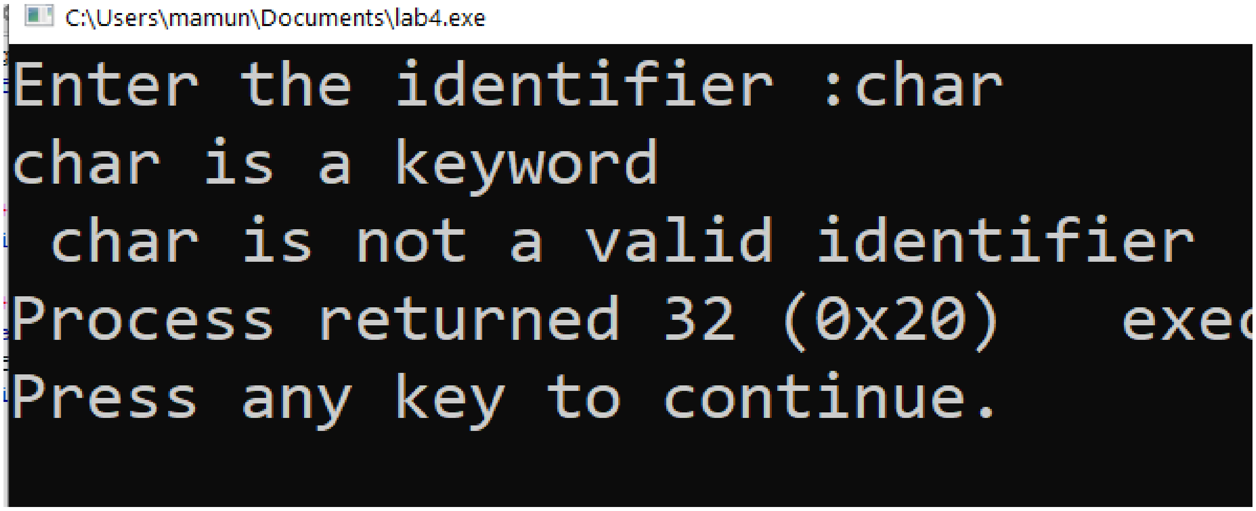
Output:

```
aaaabbbbb is accepted under rule 'a*b+'

Enter a string: cdgs

cdgs is not recognized
```

4. Question : Write a C program to validate identifiers and detect if the identifier is C keywords.

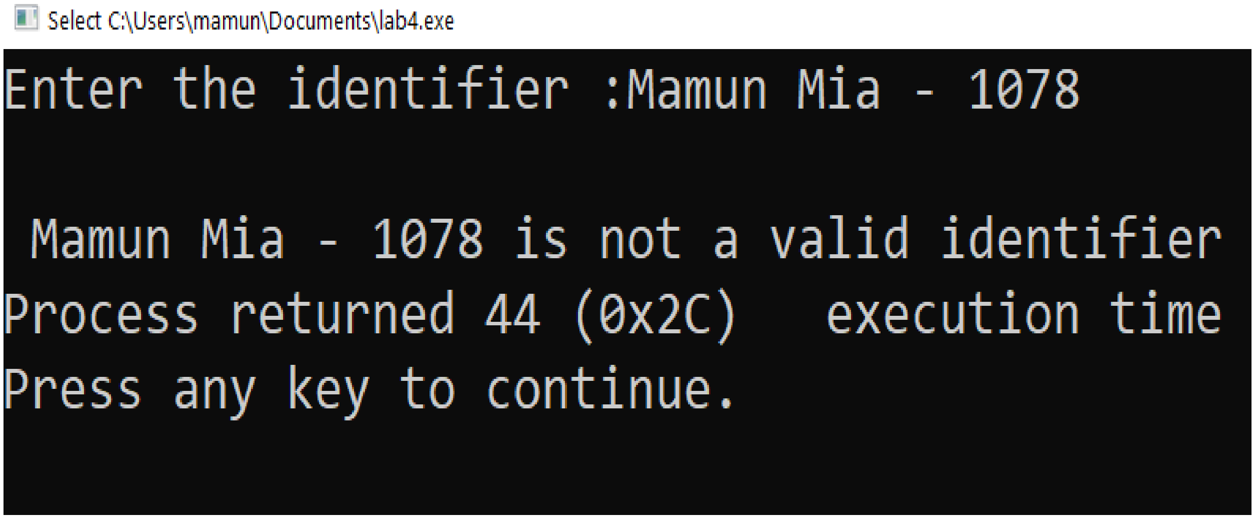
```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int i=0, flag=0;
    char keyword[32] [10]={"auto", "break", "case", "char", "const", "continue",
"default","do", "double",
"else","enum","extern","float","for","goto","if","int","long","register","return","sho
rt","signed","sizeof","static","struct","switch","typeof","union","unsigned","void","
volatile","while"};
    int a[10];
    printf("Enter the identifier :");
    gets(a);
    for(i=0;i<10;i++){
        if((strcmp(keyword[i],a)==0)){
            flag=1;
        }
    }
    if(flag==1){
        printf("%s is a keyword",a);
    }
    else
    flag=0;
    if((a[0]=="_") ||(isalpha(a[0]!=0))){
        for (i=1;a[i]!= '\0';i++){
            if((isalnum(a[i])==0) && (a[i]!='_')){
                flag=1;
            }
        }
    }
    else{
        flag=1;
    }
    if(flag==0){
        printf("\n %s is an identifier.",a);
    }
    else{
        printf("\n %s is not a valid identifier",a);
    }
    return 0;
}
```

Answer is :-



C:\Users\mamun\Documents\lab4.exe

```
Enter the identifier :char
char is a keyword
 char is not a valid identifier
Process returned 32 (0x20)   exec
Press any key to continue.
```



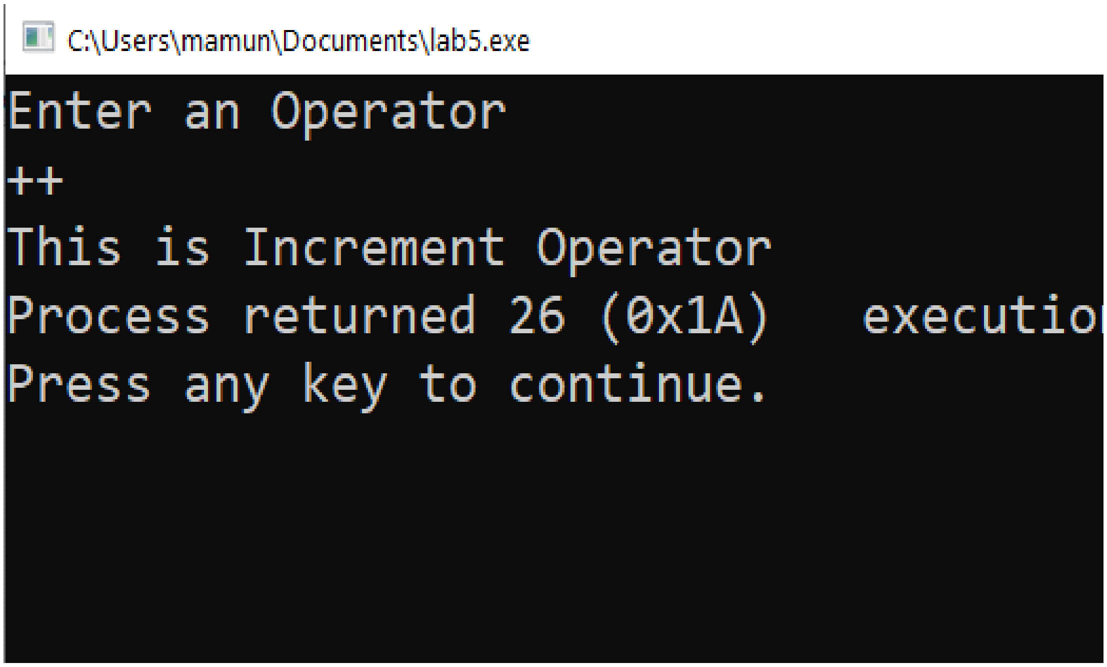
Select C:\Users\mamun\Documents\lab4.exe

```
Enter the identifier :Mamun Mia - 1078

 Mamun Mia - 1078 is not a valid identifier
Process returned 44 (0x2C)   execution time
Press any key to continue.
```

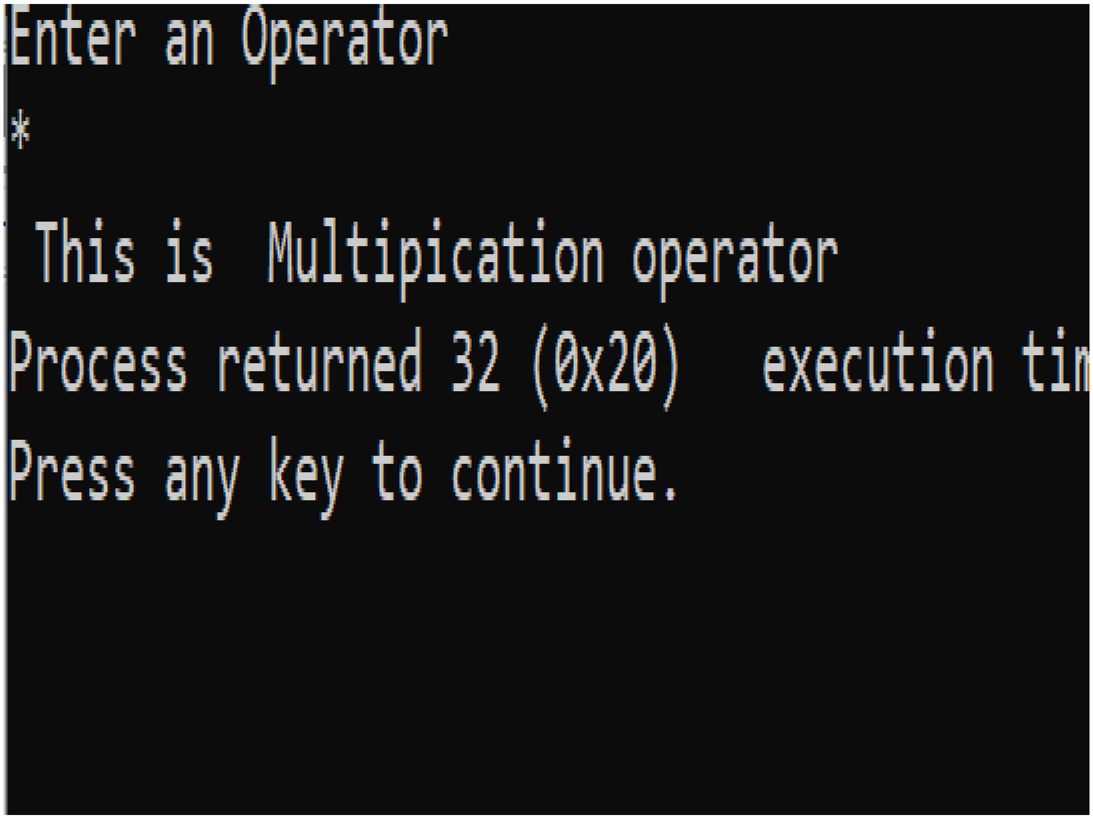
5. Question : Write a C program to simulate lexical analyzer for validating operators.

```
#include<stdio.h>
#include<string.h>
void main(){
    char a[5];
    printf("Enter an Operator\n");
    gets(a);
    int len= strlen(a);
    switch(a[0]){
    case '>':
        if(len==1) printf("This is greater than");
        else if(a[1]=='=') printf("This is greater equal");
        else printf("This is Invalid operator");
        break;
    case '<':
        if(len==1) printf("This is less than");
        else if(a[1]=='=') printf(" This is less< equal");
        else printf("\noperator is invalid");
        break;
    case '=':
        if(len==1) printf("This is assignment Operator");
        else if(a[1]=='=') printf("This is equal to");
        else printf("This is operator is invalid");
        break;
    case '!':
        if(len==1) printf("This is negation Operator");
        else if(a[1]=='=') printf("This is not equal");
        else printf("This is operator is invalid");
        break;
    case '&':
        if(len==1) printf("This is Bit wise AND");
        else if(a[1]=='&') printf("This is logical AND Operator");
        else printf("This is operator is invalid");
break;
    case '|':
        if(len==1) printf("This is Bit wise OR");
        else if(a[1]=='|') printf(" This is logical OR");
        else printf("This is operator is invalid");
        break;
    case '+':
        if(len==1) printf("\nAddition");
        else if(a[1]=='+') printf("This is Increment Operator");
        else printf("This is operator is invalid");
        break;
    case '-':
        if(len==1) printf("This is Subtraction");
        else if(a[1]=='-') printf("This is Decrement Operator");
        else printf("This is  operator is invalid");
        break;
    case '*': printf(" This is  Multipication operator");
        break;
    case '/':
        printf("This is  Division operator");
        break;
    case '%':
        printf(" This is Modulas operator");
        break;
    default:
        printf("This is  Invalid Operator");
    }
    return 0;
}
```

Answer is :-



```
Enter an Operator
++
This is Increment Operator
Process returned 26 (0x1A)   executio
Press any key to continue.
```



```
Enter an Operator
*

 This is  Multipication operator
Process returned 32 (0x20)   execution tim
Press any key to continue.
```