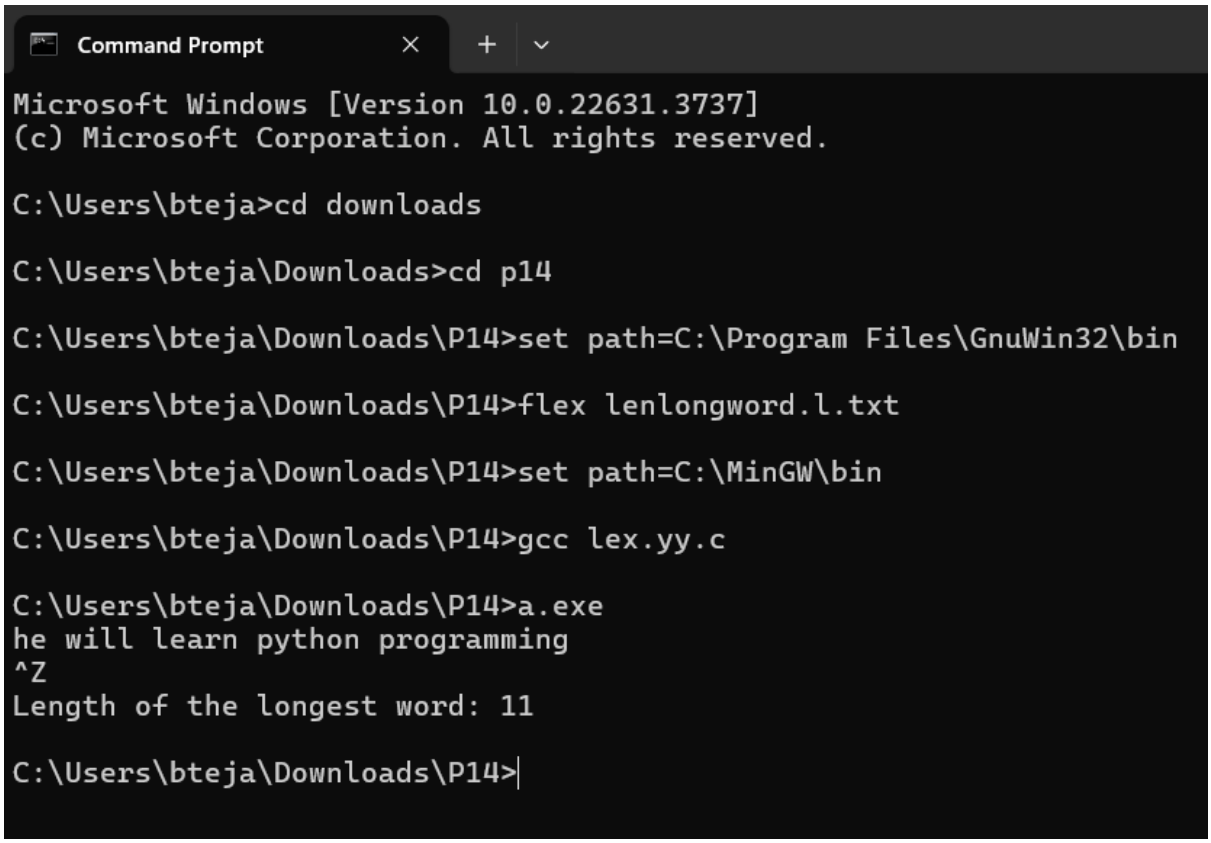
9.Write a LEX program to find the length of the longest word

PROGRAM:

```
%{
#include <stdio.h>
#include <string.h>

int max_word_length = 0;

%}

%%

[A-Za-z]+       {
                  int len = strlen(yytext);
                  if (len > max_word_length) {
                     max_word_length = len;
                  }
                }
[ \t\n]+        { /* ignore whitespace */ }
.               { /* ignore other characters */ }
-
%%

int yywrap() {
   return 1;
}

int main() {
   yylex();
   printf("Length of the longest word: %d\n", max_word_length);
   return 0;
}
```

OUTPUT:

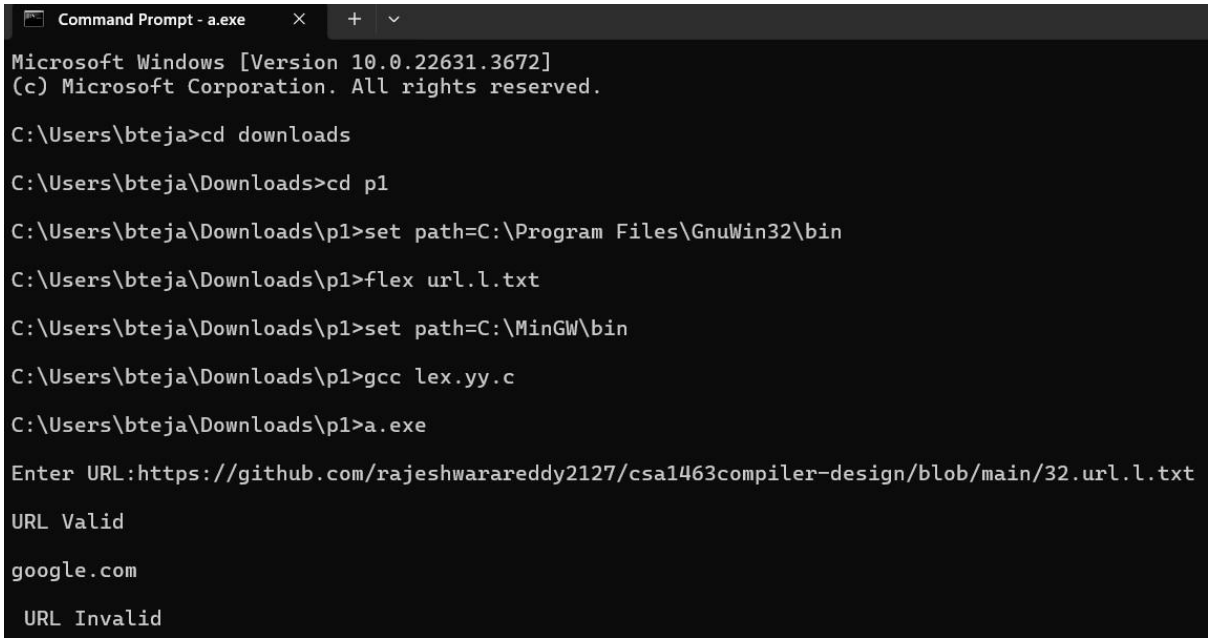


10. **A networking company wants to validate the URL for their clients. Write a LEX program to implement the same**

PROGRAM:

```
%{
%}
%%
((http)|(ftp))s?:\/\/[a-zA-Z0-9](.[a-z])+(.[a-zA-Z0-9+=?]*)* {printf("\nURL
Valid\n");}
.+ {printf("\n URL Invalid\n");}
%%
int yywrap(){}
void main()
{
printf("\nEnter URL:");
yylex();
printf("\n");
}
```

OUTPUT:

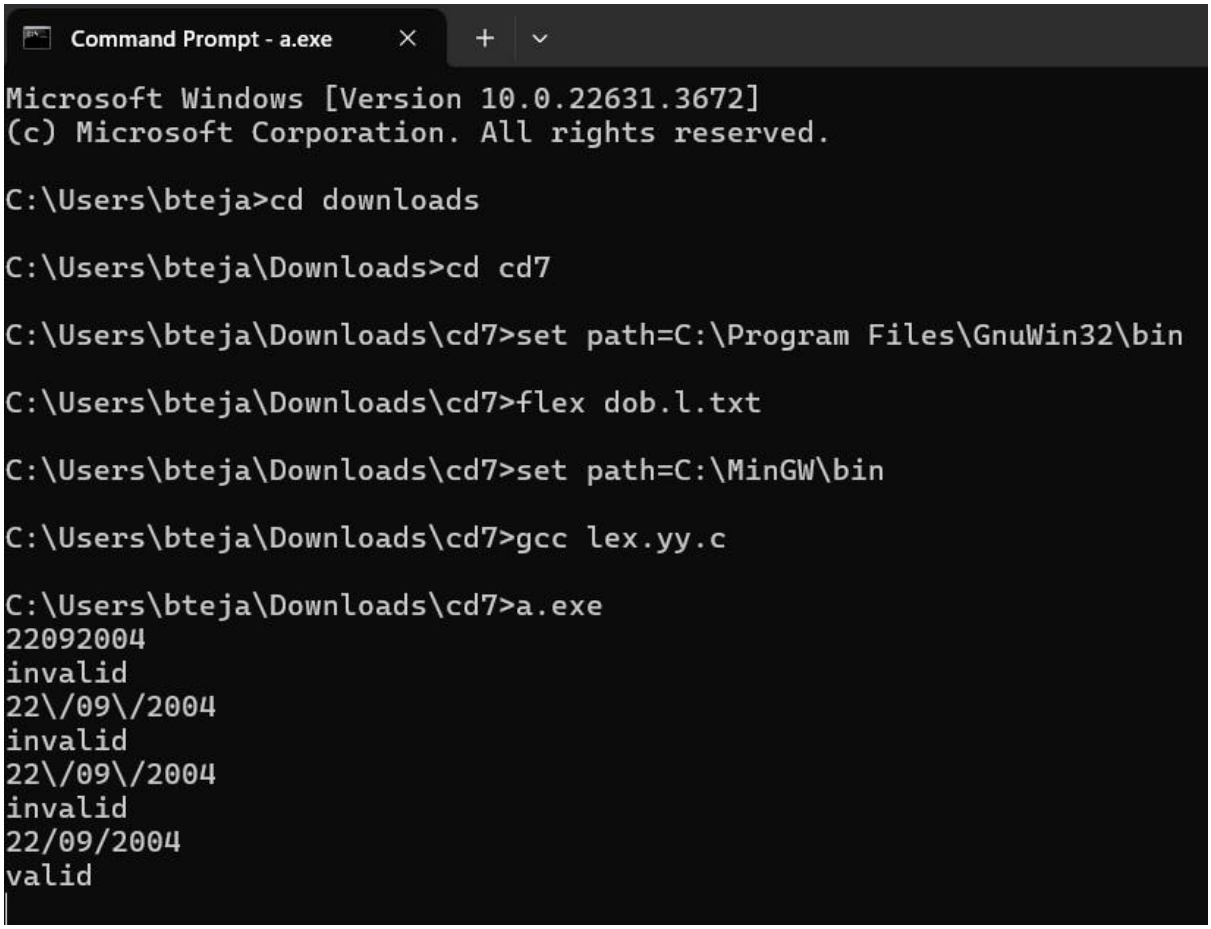


11.School management wants to validate DOB of all students. Write a LEX program to implement it

PROGRAM:

```
%{
#include<stdio.h>
%}
%%
[0-9][0-9]\/[0-1][0-9]\/[1-2][0-9]{3} {printf("valid");}
.+ {printf("invalid");}
%%
int yywrap(){}
int main()
{
yylex();
}
```

OUTPUT:

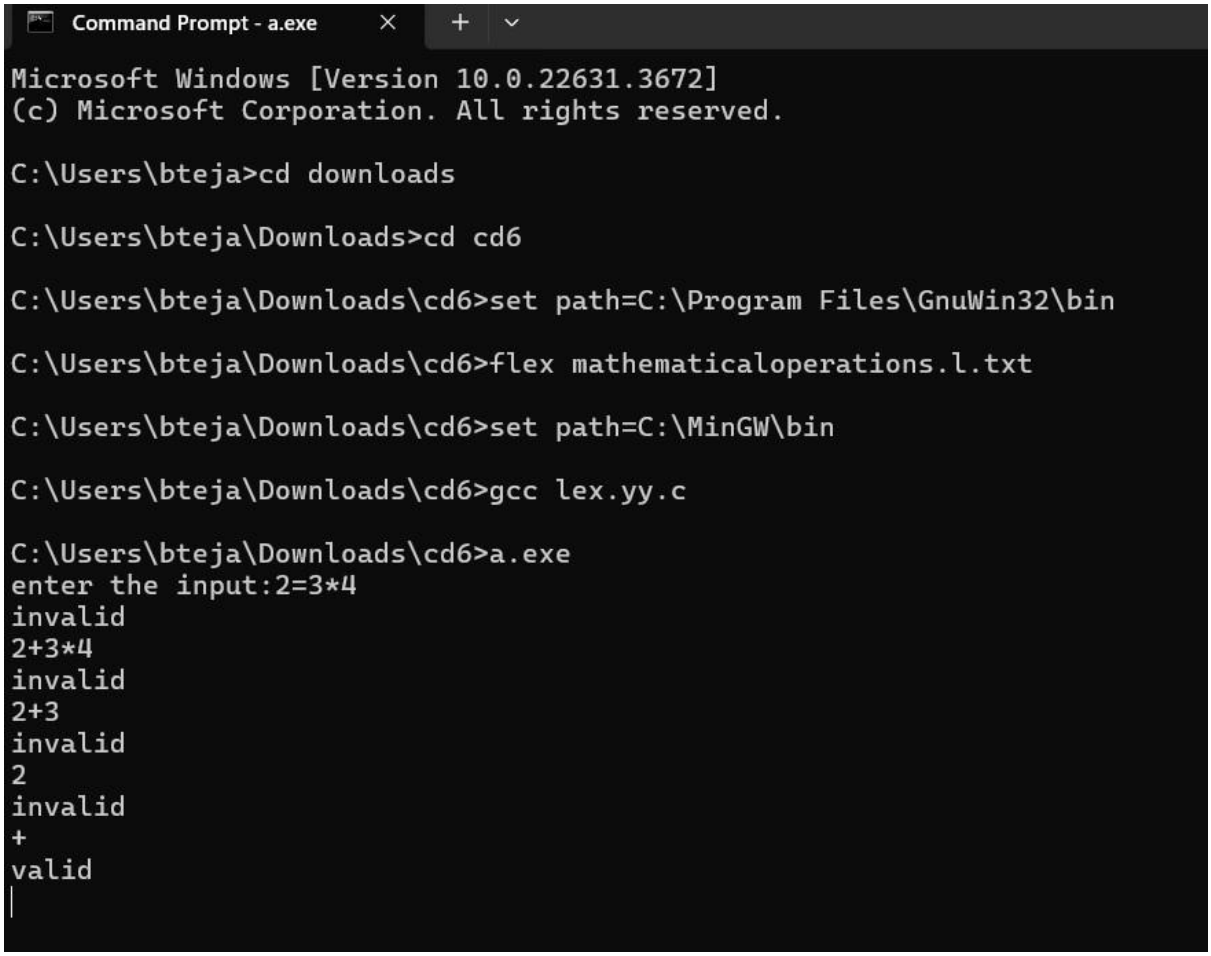


12.A School student was asked to do basic mathematical operations. Implement a LEX program to implement the same

PROGRAM:

```
%{
#include<stdio.h>
%}
%%
"="|"+"|"-"|"/"|"*" {printf("valid");}
.+ {printf("invalid");}
%%
int yywrap(){}
int main()
{
printf("enter the input:");
yylex();
return 0;
}
```

OUTPUT:

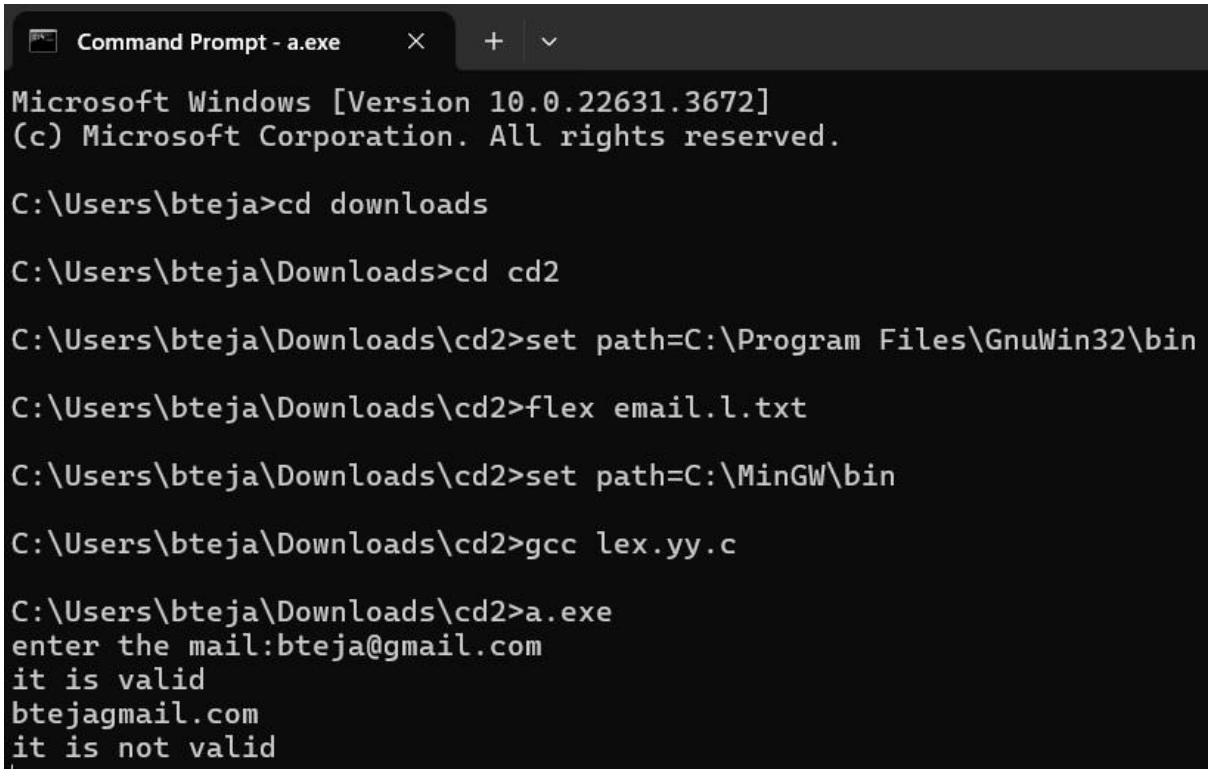


12. **Write a LEX Program to check the email address is valid or not**

    PROGRAM:

```
%{
%}
%%
[a-z.0-9_]+@[a-z]+".com"|".in" {printf("it is valid");}
.+ {printf("it is not valid");}
%%
int yywrap(void)
{}
int main()
{
printf("enter the mail:");
yylex();
}
```

    OUTPUT:



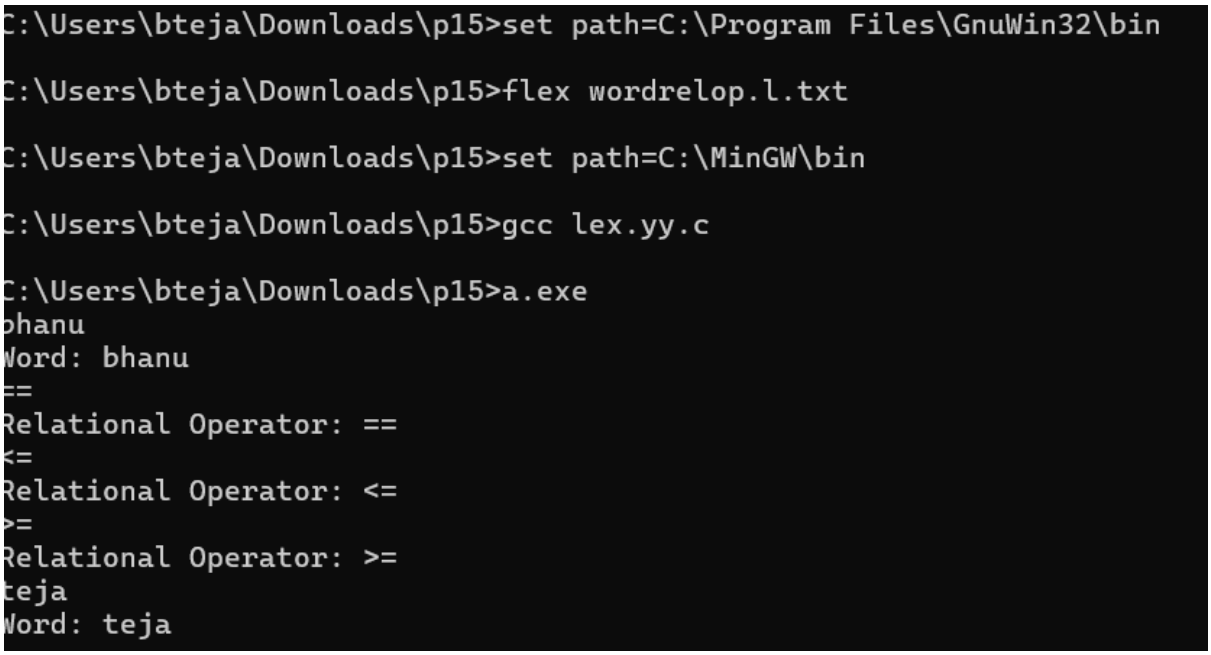
13. Write a LEX program to recognize a word and relational operator?

PROGRAM:

```
%{
#include <stdio.h>
#include <ctype.h>
%}
%%
[A-Za-z]+      { printf("Word: %s\n", yytext); }
"=="           { printf("Relational Operator: %s\n", yytext); }
"!="           { printf("Relational Operator: %s\n", yytext); }
"<="           { printf("Relational Operator: %s\n", yytext); }
">="           { printf("Relational Operator: %s\n", yytext); }
"<"            { printf("Relational Operator: %s\n", yytext); }
">"            { printf("Relational Operator: %s\n", yytext); }
```

```
[ \t\n]+       { /* ignore whitespace */ }

.              { /* ignore other characters */ }

%%

int yywrap() {

    return 1;

}

int main() {

    yylex();

    return 0;

}
```

OUTPUT:

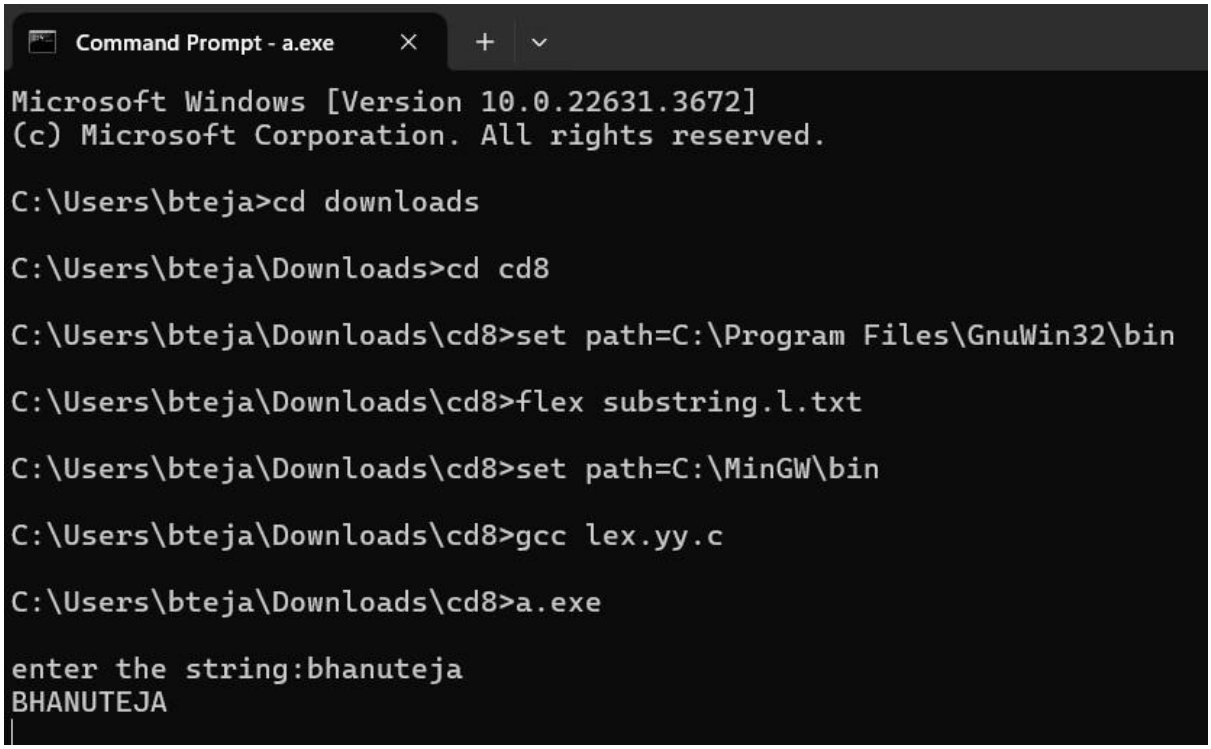


14. Write a LEX Program to convert the substring abc to ABC from the given input string

PROGRAM:

```
%{
%}
%%
[a-z] {printf("%c",yytext[0]-32);}
. {}
%%
int yywrap(void){}
int main()
{
printf("\nenter the string:");
yylex();
}
```

OUTPUT:



15. Write a lex program for addline number?

PROGRAM:

```
%{
int ln = 0;
%}
%%
.* { ln++; fprintf(yyout,"\n%d:%s",ln,yytext); }
%%
int yywrap() {
}
int main() {
yyin = fopen("simple.txt.txt", "r");
yyout = fopen("out.txt", "w");
yylex();
}
```

OUTPUT:

```
1:#include<stdio.h>
2:int main()
3:{
4:printf("hello world");
5:}
```