

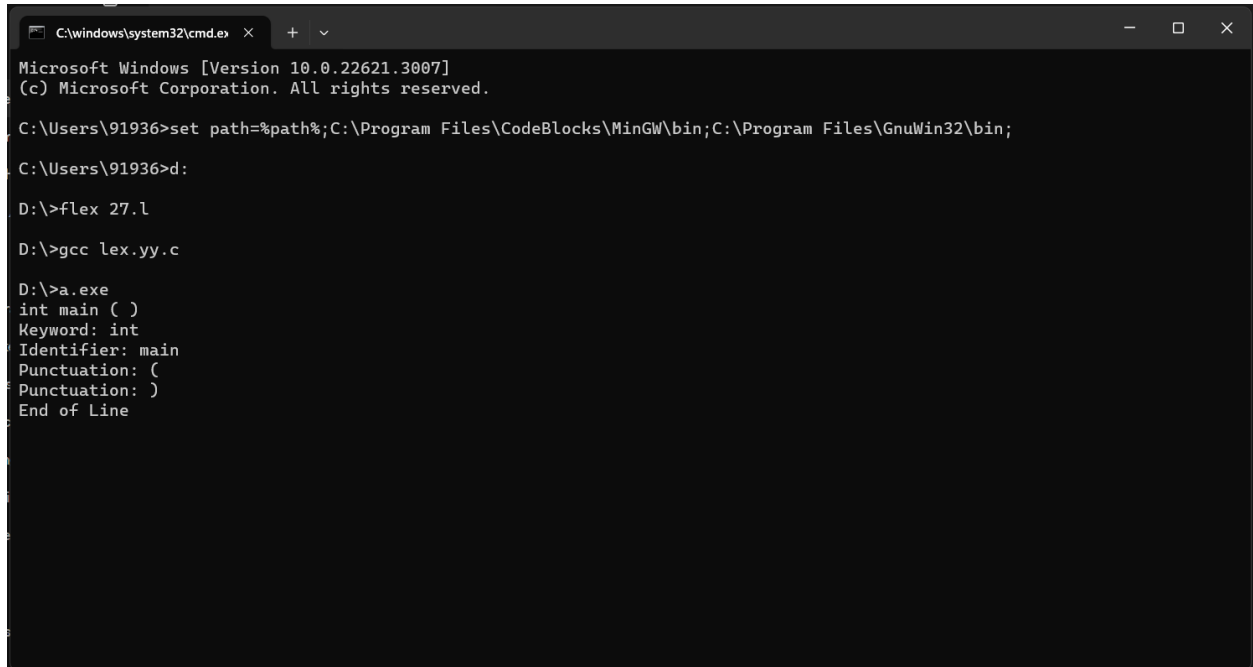
27. Implement Lexical Analyzer using LEX or FLEX (Fast Lexical Analyzer). The program should separate the tokens in the given C program and display with appropriate caption.

```
%{  
  
#include <stdio.h>  
  
#include <string.h>  
  
%}  
  
%%  
  
int|char|float|double|void    { printf("Keyword: %s\n", yytext); }  
  
[0-9]+                        { printf("Integer Literal: %s\n", yytext); }  
  
\"([^\"]|\\.)*\">  
\'([^\']|\\.)*\'  
{ printf("String Literal: %s\n", yytext); }  
{ printf("Character Literal: %s\n", yytext); }  
  
[_a-zA-Z][_a-zA-Z0-9]*        { printf("Identifier: %s\n", yytext); }  
  
[+|-|*|/=|<=>]                { printf("Operator: %s\n", yytext); }  
  
[;,() ]                       { printf("Punctuation: %s\n", yytext); }  
  
[ \t ]                        ; // skip whitespaces  
  
\\n                            { printf("End of Line\n"); }  
  
.  
{ printf("Other: %s\n", yytext); }  
  
%%  
  
int main() {  
  
    yylex();
```

```
return 0;
```

```
}
```

```
int yywrap() { return 1; }
```



```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91936>set path=%path%;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\GnuWin32\bin;

C:\Users\91936>d:

D:\>flex 27.1

D:\>gcc lex.yy.c

D:\>a.exe
int main ( )
Keyword: int
Identifier: main
Punctuation: (
Punctuation: )
End of Line
```