

GitHub repository:

<https://github.com/Lois-Beltech/Lois-Beltech-Formal-Languages-and-Compiler-Design/tree/main/Lab8>

Documentation Lab 8 – FLEX

1. Demo
2. Flex specification file

1. Demo
  - first step will be to run & compile

```
D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8>flex scanner.lxi

D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8>dir
Volume in drive D has no label.
Volume Serial Number is 1AB1-8491

Directory of D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8

06.01.2024  23:15  <DIR>          .
06.01.2024  23:15  <DIR>          ..
18.12.2023  11:32          40.737 lab8Link.pdf
08.01.2024  08:43          41.769 lex.yy.c
18.12.2023  09:21           181 p1.txt
18.12.2023  09:21           183 p1err.txt
18.12.2023  09:21           437 p2.txt
18.12.2023  09:21           222 p3.txt
06.01.2024  19:33           1.202 scanner.lxi
               8 File(s)          150.345 bytes
               2 Dir(s)  252.144.549.888 bytes free

D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8>gcc lex.yy.c

D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8>dir
Volume in drive D has no label.
Volume Serial Number is 1AB1-8491

Directory of D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8

08.01.2024  08:44  <DIR>          .
08.01.2024  08:44  <DIR>          ..
08.01.2024  08:44          65.614 a.exe
18.12.2023  11:32          40.737 lab8Link.pdf
08.01.2024  08:43          41.769 lex.yy.c
18.12.2023  09:21           181 p1.txt
18.12.2023  09:21           183 p1err.txt
18.12.2023  09:21           437 p2.txt
18.12.2023  09:21           222 p3.txt
06.01.2024  19:33           1.202 scanner.lxi
               8 File(s)          150.345 bytes
               2 Dir(s)  252.144.549.888 bytes free
```

-an executable was created after the second command, so now we can run the program a.exe

-we have 4 eprograms (p1.txt, p2.txt, p3.txt and p1err.txt)

-I'm going to run the program for p1err.txt, using the following command, & having the output:

```
D:\GitHub\Formal Languages and Compiler Design\Formal-Languges-and-Compiler-Design\Lab8>a.exe plerr.txt
prog - reserved word
{ - separator
a - identifier
( - separator
int - reserved word
) - separator
, - separator
Error at token 2b at line 2
```

## 2. Flex specification file

```
%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int lines = 1;

}%

%option noyywrap

%option caseless

DIGIT [0-9]
NON_ZERO_DIGIT [1-9]
INT_CONSTANT [+]?{NON_ZERO_DIGIT}{DIGIT}*|0
LETTER [a-zA-Z_]
SPECIAL_CHAR [ ?:*^\^+=.!]
STRING_CONSTANT (\("{LETTER}|{DIGIT}|{SPECIAL_CHAR})*\")
IDENTIFIER {LETTER}({LETTER}|{DIGIT})*
BAD_IDENTIFIER ({DIGIT})+({LETTER})+({LETTER}|{DIGIT})*

%%
```

```
"prog"|"int"|"str"|"char"|"read"|"if"|"else"|"print"|"do"|"while"|"arr"|"const"|"fun" {printf("%s - reserved word\n", yytext);}
```

```
"+"|"-"|"*"|"/"|"<"|<="|"="|>="|"=="|"<>"|>"|"sqrt" printf("%s - operator\n", yytext);
```

```
{IDENTIFIER} {printf("%s - identifier\n", yytext);}
```

```
{BAD_IDENTIFIER} {printf("Error at token %s at line %d\n", yytext, lines); exit(1);}
```

```
{INT_CONSTANT} {printf("%s - integer constant\n", yytext);}
```

```
{STRING_CONSTANT} {printf("%s - string constant\n", yytext);}
```

```
"["|"]"|";"|"("|")"|"{"|"}"|"'," printf("%s - separator\n", yytext);
```

```
[ \t]+ {}
```

```
[\n]+ {++lines;}
```

```
. {printf("Error at token %s at line %d\n", yytext, lines); exit(1);}
```

```
%%
```

```
int main(int argc, char** argv) {
```

```
    if (argc > 1)
```

```
        yyin = fopen(argv[1], "r");
```

```
    else
```

```
        yyin = stdin;
```

```
yylex();  
}
```