GitHub link: https://github.com/edwardinio18/LFTC/tree/main/Labs/Lab8

Lex 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 [stergete]?{NON_ZERO_DIGIT}{DIGIT}*|0
LETTER [a-zA-Z_]
SIGNS [ _.,:;]
STRING_CONSTANT \"[^\"]*\"
IDENTIFIER (€{LETTER})({LETTER}|{DIGIT}|_)*
BAD_IDENTIFIER (^€)({LETTER}|{DIGIT}|_)*

%%

"cvnou"|"daca"|"altfel"|"cattimp"|"fa"|"pentrufiecare"|"oprestete"|"continu
a"|"scrie"|"citeste"|"in"|"prog"|"de"|"atunci"|"incepe"|"termina"|"intreg"|
"real"|"sfoara"|"sir"|"constanta"|"functie"|"intoarce"|"advfals"|"caracter"
|"radical"|"sisi"|"sausau" {printf("RESERVED WORD: %s\n", yytext);}

"adunate"|"stergete"|"oriori"|"orioriinvers"|"estiegal"|"verificaegal"|"ver
ificanuegal"|"maimare"|"maimic"|"maimareegal"|"maimicegal"|"lasuta"
{printf("OPERATOR: %s\n", yytext);}

"spatiu"|"sarilinia"|"gatalinia"|";"|":"|"."|","|"("|")"|"{"|"}"|"["|"]"|"_
" {printf("SEPARATORS: %s\n", yytext);}

{IDENTIFIER} {printf("IDENTIFIER: %s\n", yytext);}

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

{INT_CONSTANT} {printf("INTEGER CONSTANT: %s\n", yytext);}

{STRING_CONSTANT} {printf("STRING CONSTANT: %s\n", yytext);}

[ \t]+ {}

"//"(.)*[\n]+ {++lines;}

[\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();
}
```

Demo

1. Install flex (I'm using Mac so I will use brew):
```
brew install flex
```
2. Generate the lexer code:
```
flex lang.lxi
```
3. Compile the generated C code:
```
gcc -o lang lex.yy.c -ll
```
4. Run the lexer:
```
./lang p1.txt
```
5. Output of p1.txt:
```
RESERVED WORD: cvnou
IDENTIFIER: €a
SEPARATORS: :
RESERVED WORD: intreg
SEPARATORS: ;
RESERVED WORD: cvnou
IDENTIFIER: €b
SEPARATORS: :
RESERVED WORD: intreg
SEPARATORS: ;
RESERVED WORD: cvnou
IDENTIFIER: €c
SEPARATORS: :
RESERVED WORD: intreg
SEPARATORS: ;
RESERVED WORD: citeste
SEPARATORS: (
IDENTIFIER: €a
SEPARATORS: )
SEPARATORS: ;
RESERVED WORD: citeste
SEPARATORS: (
IDENTIFIER: €b
SEPARATORS: )
SEPARATORS: ;
RESERVED WORD: citeste
SEPARATORS: (
IDENTIFIER: €c
SEPARATORS: )
SEPARATORS: ;
IDENTIFIER: €smallest_int
SEPARATORS: :
RESERVED WORD: intreg
SEPARATORS: ;
IDENTIFIER: €smallest_int
OPERATOR: estiegal
IDENTIFIER: €a
SEPARATORS: ;
RESERVED WORD: daca
SEPARATORS: (
IDENTIFIER: €smallest_int
OPERATOR: maimare
IDENTIFIER: €b
SEPARATORS: )
```

```
RESERVED WORD: atunci
SEPARATORS: {
IDENTIFIER: €smallest_int
OPERATOR: estiegal
IDENTIFIER: €b
SEPARATORS: ;
SEPARATORS: }
RESERVED WORD: daca
SEPARATORS: (
IDENTIFIER: €smallest_int
OPERATOR: maimare
IDENTIFIER: €c
SEPARATORS: )
RESERVED WORD: atunci
SEPARATORS: {
IDENTIFIER: €smallest_int
OPERATOR: estiegal
IDENTIFIER: €c
SEPARATORS: ;
SEPARATORS: }
RESERVED WORD: scrie
SEPARATORS: (
IDENTIFIER: €smallest_int
SEPARATORS: )
SEPARATORS: ;
```