



UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS FLORESTAL
CURSO CIÊNCIA DA COMPUTAÇÃO

LUCAS GABRIEL 03493

COMPILADORES
PORTFÓLIO 1

Florestal, MG

2021

Lex1

Definições

```
11 /* definicoes regulares */
12 letter [A-Za-z]
13 upperCaseLetter [A-Z]
14 lowerCaseLetter [a-z]
15 digit [0-9]
16 separation (" "){2,}
17 word ({letter}+)
18 positiveNumber ({digit}+)
19 negativeNumber ([-]{digit}+)
20 decimalNumber (([-]{digit}+[.]{digit}+)|({digit}+[.]{digit}+))
21 phoneNumber ({digit}{4}[-]{digit}{4})
22 fullName (({upperCaseLetter})(lowerCaseLetter}{+})([ \n\t\r]{1}|$)){3,4}
23 carPlate ({upperCaseLetter}{3}[-]{digit}{4})
24 delim [ \t\n]
25 ws {delim}+
```

- + → indica uma ou mais repetições do grupo
- * → indica 0 ou mais repetições
- {n} → repetição n vezes
- {n,n2} → n <= número de repetições <= n2

Na implementação do arquivo Lex1 a maior dificuldade foi realizar a captura do nome, uma vez que as demais operações consistiam em combinações fixas, porém no nome tem-se a possibilidade de se haver 3 ou 4 nomes e mesmo que seja uma implementação fácil, ela pode se complicar caso se opte por não utilizar repetições cruas como {palavra}{espaço}{palavra}...

Agora quebraremos a definição fullName:

```
fullName (({upperCaseLetter})(lowerCaseLetter}{+})([ \n\t\r]{1}|$)){3,4}
```

{upperCaseLetter} → faz com que iniciemos a palavra com uma letra maiúscula

{lowerCaseLetter}+ → faz com que tenhamos uma ou mais letras minúsculas após a primeira letra maiúscula

[\n\t\r]{1}\$ → Garante que tenhamos um espaço ou quebra de linha e caso isso não ocorra isso significará que temos um final de string (\$)

{3,4} → Indica que só serão aceitas como fullName as strings que estiverem de acordo com a regra do grupo de 3 a 4 vezes consecutivas, isso garante que tenhamos um mínimo de três e máximo de 4 nomes.

```
34 {positiveNumber} {printf("Foi encontrado um numero inteiro positivo. LEXEMA: %s\n", yytext);}
35 {negativeNumber} {printf("Foi encontrado um numero inteiro negativo. LEXEMA: %s\n", yytext);}
36 {phoneNumber} {printf("Foi encontrado um telefone. LEXEMA: %s\n", yytext);}
37 {fullName} {printf("Foi encontrado um nome completo. LEXEMA: %s\n", yytext);}
38 {decimalNumber} {printf("Foi encontrado um numero com parte decimal. LEXEMA: %s\n", yytext);}
39 {word} {printf("Foi encontrado uma palavra. LEXEMA: %s\n", yytext);}
40 {carPlate} {printf("Foi encontrado uma placa. LEXEMA: %s\n", yytext);}
41 {ws} { /*nenhuma acao e nenhum retorno*/ }
```

Arquivo de exemplo

```
1 875878 -3355456 abc5464      abc-5464 ABC-5464      453-2345 9486-0847
2 Daniel Mendes Barbosa 32.345  Palavra Qualquer 3567-3224
3 Daniel Mendes Barbosa Daniel Mendes Barbosa  Menezes200 LucasGabrielBarbosa
```

Output do arquivo

```
luke@luke-VirtualBox: ~/Desktop/Compiladores/Atividade 1
luke@luke-VirtualBox:~/Desktop/Compiladores/Atividade 1$ ./a.out < input-example.txt
Foi encontrado um numero inteiro positivo. LEXEMA: 875878
Foi encontrado um numero inteiro negativo. LEXEMA:: -3355456
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um numero inteiro positivo. LEXEMA: 5464
Foi encontrado uma palavra. LEXEMA: abc
Foi encontrado um numero inteiro negativo. LEXEMA:: -5464
Foi encontrado uma placa. LEXEMA: ABC-5464
Foi encontrado um numero inteiro positivo. LEXEMA: 453
Foi encontrado um numero inteiro negativo. LEXEMA:: -2345
Foi encontrado um telefone. LEXEMA:: 9486-0847
Foi encontrado um nome completo. LEXEMA:: Daniel Mendes Barbosa
Foi encontrado um numero com parte decimal. LEXEMA: 32.345
Foi encontrado uma palavra. LEXEMA: Palavra
Foi encontrado uma palavra. LEXEMA: Qualquer
Foi encontrado um telefone. LEXEMA:: 3567-3224
Foi encontrado um nome completo. LEXEMA:: Daniel Mendes Barbosa Daniel
Foi encontrado uma palavra. LEXEMA: Mendes
Foi encontrado uma palavra. LEXEMA: Barbosa
Foi encontrado uma palavra. LEXEMA: Menezes
Foi encontrado um numero inteiro positivo. LEXEMA: 200
Foi encontrado uma palavra. LEXEMA: LucasGabrielBarbosa
luke@luke-VirtualBox:~/Desktop/Compiladores/Atividade 1$
```

Lex2

Definição

Como proposta para a parte 2 foi implementado a identificação de CPFs CNPJs emails e moedas, podendo variar entre dólar, real, euro e libra, abaixo seguem os prints das implementações e dos exemplos.

```
11 /* definicoes regulares */
12 letter [A-Za-z]
13 upperCaseLetter [A-Z]
14 lowerCaseLetter [a-z]
15 letterOrNumber [a-zA-Z0-9]
16 digit [0-9]
17 cpf ((({digit}{3}[.]){2}){digit}{3}[-]{digit}{2})
18 cnpj ({digit}{2}[.]{digit}{3}[.]{digit}{3}/]{digit}{4}[-]{digit}{2})
19 email ({letterOrNumber}+("@"){letterOrNumber}+([.][a-zA-z]+){0,2})
20 dolar (("€"){digit}+[.]{digit}{1,2})
21 real (("R$"){digit}+[.]{digit}{1,2})
22 euro (("€"){digit}+[.]{digit}{1,2})
23 libra (("£"){digit}+[.]{digit}{1,2})
24
25 delim      [ \t\n]
26 ws         {delim}+
```

```
37 {cpf} {printf("Foi encontrado um cpf. LEXEMA:: %s\n", yytext);}
38 {cnpj} {printf("Foi encontrado um cnpj. LEXEMA:: %s\n", yytext);}
39 {dolar} {printf("Foi encontrado um valor em dolar. LEXEMA:: %s\n", yytext);}
40 {real} {printf("Foi encontrado um valor em real. LEXEMA:: %s\n", yytext);}
41 {euro} {printf("Foi encontrado um valor em euro. LEXEMA:: %s\n", yytext);}
42 {libra} {printf("Foi encontrado um valor em libra. LEXEMA:: %s\n", yytext);}
43 {email} {printf("Foi encontrado um email. LEXEMA:: %s\n", yytext);}
44 {ws}      {/*nenhuma acao e nenhum retorno*/}
```

Input de teste

```
1 06.947.284/0001-04    642.748.951-41 email@dominio $42.75 danielmendes@ufv.br
2 £75.84 R$8476.42 €870.50 00.000.000/0001-91 06.947.284/0001-04 teste@testado.testei.br
3
```

Output do arquivo de teste

```
luke@luke-VirtualBox:~/Desktop/Compiladores/Atividade 2$ ./a.out < input-atividade2.txt
Foi encontrado um cnpj. LEXEMA:: 06.947.284/0001-04
Foi encontrado um cpf. LEXEMA:: 642.748.951-41
Foi encontrado um email. LEXEMA:: email@dominio
Foi encontrado um valor em dolar. LEXEMA:: $42.75
Foi encontrado um email. LEXEMA:: danielmendes@ufv.br
Foi encontrado um valor em libra. LEXEMA:: £75.84
Foi encontrado um valor em real. LEXEMA:: R$8476.42
Foi encontrado um valor em euro. LEXEMA:: €870.50
Foi encontrado um cnpj. LEXEMA:: 00.000.000/0001-91
Foi encontrado um cnpj. LEXEMA:: 06.947.284/0001-04
Foi encontrado um email. LEXEMA:: teste@testado.testei.br
luke@luke-VirtualBox:~/Desktop/Compiladores/Atividade 2$
```