

**EXA869 MI Proc. de Linguagem de Programação**  
**Problema 1: Quem fez fez quem não fez não faz mais!**  
 Parte 1.

**Cronograma**

Sessão	Data	Assunto
1	26/fev	Apresentação Problema 1
2	28/fev	Problema 1
	05/mar	<b>Feriado Carnaval</b>
3	07/mar	Problema 1
4	12/mar	Problema 1
5	14/mar	Problema 1
6	19/mar	Problema 1
7	21/mar	Problema 1

Mais um semestre letivo se inicia e os alunos precisam decidir quais disciplinas cursarão. João e Maria são alunos de Engenharia de Computação de uma conceituada universidade do interior da Bahia, e João comentou com Maria que iria cursar Linguagens Formais e Compiladores. E sem pestanejar, Maria disse: “Você está maluco! Todo mundo diz que esta disciplina é um saco!”. João disse para Maria: “Em primeiro lugar eu não sou todo mundo. Em segundo lugar, eu dei uma pesquisada na internet sobre o assunto e não achei tão chato assim, pelo contrário, achei interessante. Você já se perguntou como os nossos algoritmos são executados por um computador? Você já parou para pensar como é construída uma linguagem de programação?”. Maria pensou por alguns segundos e disse: “É... verdade... nunca parei para pensar nisso!”. Daí João disse: “Bora fazer comigo esta disciplina?! A disciplina é chata, mas o professor é super gente fina”. Quem disse isso, perguntou Maria: “Todo mundo! Kkkk”, respondeu João. Maria disse: “Ok, vou encarar essa com você. Mas eu não sei nada sobre o assunto... o que você conseguiu estudar?”. João respondeu: “Não estudei muita coisa, mas sei que um programa chamado compilador é responsável em analisar um código fonte e a primeira etapa do compilador é fazer a análise léxica deste código. Agora, não me pergunte como isso é feito que eu não sei!”

**Problema**

O problema é o desenvolvimento de um analisador léxico para uma linguagem de programação. Para esta etapa foi disponibilizada a seguinte tabela contendo a estrutura léxica da linguagem.

Palavras reservadas	<b>variables, const, class, methods, main, return, if, else, for, read, print, void, int, float, boolean, string, true, false</b>
Identificadores	<b>letra ( letra   dígito   _ )*</b>
Números	<b>(-)? Espaço* dígito+ ( . dígito ( dígito )*)?</b>
Dígito	<b>[0-9]</b>
Letra	<b>[a-z]   [A-Z]</b>

Operadores aritméticos	<b>+ - * / ++ --</b>
Operadores relacionais	<b>!= == &lt; &lt;= &gt; &gt;= =</b>
Operadores lógicos	<b>! &amp;&amp;   </b>
Comentários	<b>// isto é um comentário de linha</b>  <b>/* isto é um comentário de bloco */</b>
Delimitadores	<b>; , . ( ) [ ] { } -&gt;</b>
Cadeia de caracteres	<b>" ( letra   dígito   símbolo ) * "</b>
Símbolo	<b>ASCII de 32 a 126 (exceto ASCII 34)</b>
Espaço	<b>ASCII 9   ASCII 32</b>

**Observações:**

1. A formação dos lexemas inicia e termina na mesma linha.
2. A entrada para este analisador é um conjunto de arquivos texto. Estes arquivos de entrada deverão estar em uma pasta na raiz do projeto chamada **files**.
3. A saída do analisador léxico deverá ser através de um conjunto de arquivos de saída (um para cada arquivo de entrada), denominados **X-saida.txt**, onde **X** é o nome do arquivo de entrada. Estes arquivos de saída deverão estar em uma pasta na raiz do projeto chamada **files**.
4. Os arquivos de saída deverão apresentar a lista de *tokens*, proveniente da análise léxica, além dos erros léxicos, caso existam. Se não houver erros, uma mensagem de sucesso deve ser gravada no arquivo de saída.

**Produto**

O analisador léxico poderá ser desenvolvido em dupla ou individualmente. **Esta dupla deverá ser a mesma para os próximos problemas.** O analisador léxico deverá ser implementado na linguagem Java ou Python. O código-fonte do analisador léxico deve ser enviado para o seu tutor, pelo classroom da disciplina, até as 23h59m do dia **23/03/2025**. Será descontado **02 (dois)** pontos por um dia de atraso. Após esse prazo, o código não será mais aceito.

**Recursos para aprendizagem**

- AHO, A. V.; SETHI, S. & ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. Rio de Janeiro: LTC, 1995.
- LOUDEN, K. C. **Compiladores – Princípios e Práticas**. São Paulo, Thomson, 2004.
- HOPCROFT, J. E. *et al.* **Introdução à Teoria dos Autômatos, Linguagens e Computação**. 1ª edição, Editora Campus, 2002.
- MENEZES, P. F. B. **Linguagens Formais e Autômatos**. 5ª edição, Editora Sagra-Luzzatto, 2005.