

	Curso	Ciência da Computação
	Atividade Acadêmica	Tradutores
	Ano/Semestre	2017/01
	Professor	Leandro Teodoro
	Data	20/04/2017
	<b>TRABALHO 1 – Tradutores</b>	

### Construa um Analisador Léxico que reconheça:

- **Variáveis ou identificadores:** este analisador léxico deve ser capaz de reconhecer variáveis declaradas e/ou utilizadas em um código fonte:

#### Exemplo:

- Trecho de código:

```
int x = 7;
```

```
int y;
```

- Tokens gerados:

```
[reserved_word, int] [id, 1] [Equal_Op, =] [num, 7]
```

```
[reserved_word, int] [id, 2]
```

- **Constantes numéricas (números inteiros):** este analisador léxico deve ser capaz de reconhecer um número inteiro qualquer e convertê-lo para os respectivos tokens:

#### Exemplo:

- Trecho de código:

```
int x = 7 + 25 * 52;
```

- Tokens gerados:

```
[reserved_word, int] [id, 1] [Equal_Op, =] [num, 7] [Arith_Op, +] [num, 25]
```

```
[Arith_Op, *] [num, 52]
```

- **Palavras reservadas:** este analisador léxico deve ser capaz de reconhecer as palavras reservadas: *do, while, if, else, for, printf, return, null, int, float, double, string, bool* e convertê-las para os respectivos tokens:

#### Exemplo:

- Trecho de código:

```
if( x == 10 )
```

- Tokens gerados:

```
[reserved_word, if] [id, 1] [Relational_Op, ==] [num, 10]
```

- **Operadores relacionais:** este analisador léxico deve ser capaz de reconhecer os operadores relacionais: *<, <=, ==, !=, >=, >* e convertê-los para os respectivos tokens:

**Exemplo:**

- Trecho de código:

```
while( x != 0)
```

- Tokens gerados:

```
[reserved_word, while] [id, 1] [Relational_Op, !=] [num, 0]
```

- **Números de ponto flutuante (números reais):** este analisador léxico deve ser capaz de reconhecer números reais quaisquer e convertê-los para os respectivos tokens:

**Exemplo:**

- Trecho de código:

```
int x = 7.15 - 2.13;
```

- Tokens gerados:

```
[reserved_word, int] [id, 1] [Equal_Op, =] [num, 7.15] [Arith_Op, -] [num, 2.13]
```

- **Remoção de espaços em branco e comentários:** este analisador léxico deve ser capaz de reconhecer espaços em branco e comentários no código fonte e removê-los.

**Exemplo:**

- Trecho de código:

```
//Comentário 1
```

```
/* Comentário 2 */
```

O trabalho pode ser realizado em grupos de até **3 alunos**, bem como deverá ser entregue pelo Moodle até o dia **20/04** e apresentado ao professor nessa mesma data. Cada grupo deve também gerar o código fonte de entrada para o Analisador Léxico que contemple toda a gramática especificada no enunciado descrito anteriormente.

O trabalho pode ser elaborado na linguagem que o grupo definir, C, C++, Java.

Um ótimo trabalho a todos!