

# Trabalhos Compiladores 2017-1 1º bim.

(individual ou 2 alunos)

## 1º bimestre: (prova 60%, trabalhos 40%)

Especificação mínima da linguagem (30%):

Prazo: 15/08/2017

- Como regra geral usar a forma de análise dos trabalhos de CLP.
- Especificar a **estrutura** geral de um programa.
- A especificação de tipos deve ser estática, com no mínimo o seguinte conjunto de tipos de dados: (devem ser especificadas quais as operações de cada tipo):
  - inteiro;
  - ponto flutuante;
  - caractere;
  - booleano;
  - cadeia de caracteres;
  - arranjos unidimensionais.
- Conjunto **mínimo** de operadores: (especificar ordem de precedência e associatividade):
  - aritméticos: para tipos numéricos
    - aditivos, multiplicativos;
    - unário negativo;
  - relacionais: para tipos numéricos, caracteres e cadeias de caracteres;
  - lógicos: para tipo booleano
    - negação, conjunção e disjunção;
  - concatenação de cadeias de caracteres.
- Instruções: (especificar natureza e formas de controle)
  - estrutura condicional de uma e duas vias;
  - estrutura iterativa com controle lógico;
  - estrutura iterativa controlada por contador (lembrar que o 'for' do C é controle lógico);
  - entrada e saída.
- Atribuição pode ser instrução ou operador (especificar).
- Funções
- Incluir os seguintes programas exemplos, que deverão ser testados usando o analisador léxico.
  - *alô mundo*;
  - série de Fibonacci, implementada em uma função, até um termo limite, listados separados por vírgula, usando iteração com controle lógico; o limite deve ser lido no programa principal;
  - *shell sort*, implementado em uma função usando iteração controlada por contador, em um arranjo cujos valores devem ser lidos no programa principal, também no programa principal listar os valores ordenados.

### Especificação dos tokens (30%):

Prazo: 22/08/2017

- Especificar a linguagem de programação em que os analisadores léxico e sintático serão implementados
- Especificar a enumeração com as categorias dos *tokens* a ser **obrigatoriamente** usada nos analisadores léxico e sintático, usando a sintaxe da linguagem escolhida para a implementação dos analisadores.
- Especificar em dois grupos **distintos**:
  - expressões regulares auxiliares, que não representam terminais da linguagem, e
  - expressões regulares dos lexemas, que representam os terminais da linguagem, especificando a categoria do *token* associado
- A especificação das expressões regulares devem seguir a simplificação do padrão EBNF ISO/IEC 14977: 1996(E) como usada em sala de aula (nomes sem espaços e sem vírgulas entre elementos das produções).

### Analizador Léxico (40%):

Prazo: 31/08/2017

- Deve ser implementado para fazer a análise “*on the fly*”, devolvendo o *token* identificado via um método/função cuja assinatura seja

Token nextToken();
--------------------

para o analisador sintático, e não como um passo em separado que faça a análise léxica toda antes da análise sintática, onde **Token** deverá ser uma **struct** ou **class** contendo os dados do token.

- Deve ser implementado um programa para teste que execute o analisador léxico, listando para cada *token*, nesta ordem, sua posição (linha e coluna) no programa exemplo, sua categoria (o número e nome associado na enumeração) e seu valor léxico, que deverá estar presente nos objetos **Token**.
- Devem ser apresentados os resultados dos testes para os três programas previamente solicitados.

**SUGERE-SE FORTEMENTE** que versões prévias sejam apresentadas ao professor para análise e discussão.