

## Especificação mínima da linguagem - PRAZO: 05/03

- A linguagem deve poder ser analisada em passo único.
- Especificar a **estrutura** geral de um programa, indicando onde podem ser declaradas funções, variáveis, instruções.
- 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.
- Especificar as constantes literais de cada tipo.
- 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: todos;
    - para tipos booleanos: igualdade e desigualdade;
  - lógicos:
    - para tipo booleano: negação, conjunção e disjunção;
  - concatenação: geram cadeias de caracteres
    - tipos caracteres e cadeias de caracteres;
    - tipos numéricos e booleanos: se concatenados a um caractere ou cadeia de caracteres, deve ser convertido para cadeia 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 com passo igual a um caso omitido;
  - entrada: deve permitir entrada de mais de uma variável em uma única instrução;
  - saída:
    - deve permitir mais de uma variável/constante literal em uma única instrução;
    - deve permitir minimamente formatação opcional de tamanho do campo e, para ponto flutuante, número de casas decimais, default 2 casas decimais; quando presente o formato deve preceder o elemento a ser impresso.
- Atribuição pode ser instrução ou operador (especificar).
- Funções.

- Incluir os seguintes programas exemplos, que deverão ser testados usando o analisador léxico:
  - alô mundo;
  - listar os elementos da série de Fibonacci até um valor limite, separados por vírgula, implementada em uma função 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 e listados no programa principal, também no programa principal listar os valores ordenados.

## **Especificação dos tokens - PRAZO: 05/03**

- 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; nomes simbólicos de até 10 caracteres.
- Especificar em dois grupos distintos:
  - tabela com nomes das expressões e as expressões regulares auxiliares, que não representam terminais da linguagem;
  - tabela com as categorias simbólicas dos tokens e as expressões regulares dos lexemas correspondentes, que representam os terminais da linguagem.
- 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); as expressões regulares auxiliares funcionam como terminais das expressões associadas aos tokens, devendo estar entre apóstrofes (') ou aspas (").

## **Analisador léxico - PRAZO: 14/03**

- O nome do programa analisado deve ser passado na linha de comando.
- 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. O formato de impressão deve seguir o modelo a seguir (em C): "[%03d. %03d] (%04d. %10s) {%s}"
  - Obs.: caso o código numérico das categorias possa ultrapassar 4 dígitos, usar o número par superior mais próximo

- Devem ser apresentados os resultados dos testes para os três programas previamente solicitados.