



Instituto Federal de Educação, Ciência e Tecnologia  
Curso: Bacharelado em Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Professor: Mário Luiz Rodrigues Oliveira  
Atividade: 2ª Trabalho Prático  
Formiga, MG, 24 de outubro de 2016

## INSTRUÇÕES:

1. Esta atividade deve, obrigatoriamente, ser resolvida individualmente.
2. Caso você ache que falta algum detalhe nas especificações, você deverá fazer as suposições que julgar necessárias e escrevê-las no seu relatório. Pode acontecer também que a descrição dessa atividade contenha dados e/ou especificações supérfluas para sua solução. Utilize sua capacidade de julgamento para separar o supérfluo do necessário.
3. Como produtos da atividade serão gerados três artefatos: códigos fontes da implementação, arquivo *MakeFile* e documentação da atividade.
4. Cada arquivo-fonte deve ter um cabeçalho constando as seguintes informações: nome autor(a), matrícula e data.
5. O arquivo contendo a documentação da atividade (relatório) deve ser devidamente identificado com o nome e matrícula do(a) autor(a) do trabalho. O arquivo contendo o relatório deve, obrigatoriamente, estar no formato **PDF**.
6. Devem ser entregues os arquivos contendo os códigos-fontes, arquivo *MakeFile* e o arquivo contendo a documentação da atividade (relatório). Compacte todos os artefatos gerados **num único arquivo no formato RAR**.
7. A atividade deve ser entregue, obrigatoriamente, via portal acadêmico acessado pela **URL: <https://meu.ifmg.edu.br/>**.
8. O prazo final para entrega desta atividade é até **23:59:00** do dia **14/11/2016**.
9. O envio é de total responsabilidade do aluno. **Não serão aceitos trabalhos enviados fora do prazo estabelecido.**
10. **Trabalhos plagiados serão desconsiderados, sendo atribuída nota 0 (zero) a todos os envolvidos.**
11. O valor desta atividade é 15 pontos.



Instituto Federal de Educação, Ciência e Tecnologia  
Curso: Bacharelado em Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Professor: Mário Luiz Rodrigues Oliveira  
Atividade: 2ª Trabalho Prático  
Formiga, MG, 24 de outubro de 2016

## 1. Árvore de Expressão, Objetivos e Descrição do Trabalho

Uma expressão aritmética pode ser representada por uma árvore binária cujos nós folhas representam os operandos e os nós internos são associados aos operadores. Esse tipo de árvore é denominado árvore de expressão. Cada nó deste tipo de árvore tem um valor associado. Se o nó é folha, o seu valor é o valor do operando e se o nó é interno, então o seu valor é definido aplicando-se sua operação sobre o valor de seus filhos.

O objetivo geral deste trabalho é o projeto e a implementação de um sistema interpretador e conversor de expressões aritméticas usando o conceito de árvore de expressão. O objetivo específico é exercitar o conceito e aplicações da estrutura de dados árvore, notadamente da estrutura de dados árvore binária.

Para tanto deve ser implementado um programa que leia expressões aritméticas na notação posfixa e gere a correspondente árvore de expressão. Deve-se, obrigatoriamente, prover a entrada de dados via teclado e também via arquivo modo texto. O programa deve permitir ao usuário escolher uma dessas opções. A entrada via arquivo modo texto deve permitir que sejam especificadas várias expressões aritméticas, uma expressão por linha. De maneira análoga, o programa deve prover duas formas de saída de dados, a saber: uma via teclado e outra via arquivo modo texto. Tal qual para a entrada de dados, a interface do programa deve permitir que o usuário escolha o formato da saída de dados.

Dada uma expressão aritmética na notação posfixa, para construir uma árvore de expressão pode-se usar o seguinte algoritmo:

1. Avalie a expressão na forma posfixa da esquerda para a direita;
2. Se o próximo símbolo da expressão é um operando, coloca-se numa pilha uma árvore composta apenas de um nó contendo esse operando, e
3. Para símbolo representando um operador binário, as duas árvores no topo da pilha correspondem aos seus operandos. As duas árvores são retiradas da pilha e cria-se uma nova árvore com o operador na raiz e as duas árvores correspondentes aos operandos como suas subárvores. Então, coloca-se a nova árvore na pilha.

Após percorrer toda a expressão, a pilha conterá uma única árvore, que é a árvore de expressão requerida.

A(s) árvore(s) de expressão(ões) gerada(s) deve(m) ser percorrida(s) de forma a realizar as seguintes funções:

1. Interpretar, avaliar e imprimir o valor da expressão aritmética representada pela árvore de expressão;



Instituto Federal de Educação, Ciência e Tecnologia  
Curso: Bacharelado em Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Professor: Mário Luiz Rodrigues Oliveira  
Atividade: 2ª Trabalho Prático  
Formiga, MG, 24 de outubro de 2016

2. Imprimir, na notação prefixa, a expressão aritmética representada pela árvore de expressão;
3. Imprimir, na notação infixa, a expressão aritmética representada pela árvore de expressão.

Podem ser assumidas as seguintes características sobre as expressões aritméticas:

1. Os operadores existentes nas expressões aritméticas serão sempre binários;
2. As expressões aritméticas serão sempre bem formadas;
3. Os operandos das expressões aritméticas são formados por um único dígito inteiro;
4. Os operadores permitidos nas expressões aritméticas são representados pelos símbolos: +, -, \* e /, representando as operações de adição, subtração, multiplicação e divisão respectivamente;
5. As precedências e associatividade dos operadores +, -, \* e / são as usualmente já conhecidas, ou seja:
  - todos os quatro operadores são associativos à esquerda
  - os operadores \* e / têm prioridade maior que os operadores + e -

Ademais, o sistema projetado deve também atender as seguintes especificações:

- o código fonte do programa deve ser portátil, ou seja, o mesmo código fonte deve ser compilado corretamente e gerar código executável para as seguintes plataformas: sistemas operacionais Windows e GNU/Linux e arquiteturas *Intel* x86/x86-64 e *AMD64*/x86-64;
- ser implementado, exclusivamente, na linguagem de programação C padrão *ANSI*;
- o código fonte do programa deve ser compatível com o compilador *GNU Compiler Collection* (GCC versão 4.6.1 ou superior). Para verificar a versão do compilador GCC instalado no seu computador use a opção `-version`. Adicionalmente o código deve ser compatível com a seguinte opção de compilação do GCC:
  1. `-Wall`: ativa todos avisos de compilação;
- utilizar a estrutura de dados árvore binária para armazenar as expressões aritméticas.

Na seção 2 são descritos os artefatos e o que deve ser entregue como produto desta atividade.

## 2. Artefatos

Esta seção descreve o que deve ser gerado como produto final do trabalho. Ao final do trabalho deve ser gerado além das implementações um relatório documentando seu sistema, com as seguintes informações:

1. introdução: descrever o problema resolvido e apresentar uma visão geral do sistema implementado;



Instituto Federal de Educação, Ciência e Tecnologia  
Curso: Bacharelado em Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Professor: Mário Luiz Rodrigues Oliveira  
Atividade: 2ª Trabalho Prático  
Formiga, MG, 24 de outubro de 2016

2. implementação: descrição sobre as decisões de projeto e implementação do programa. Essa parte da documentação deve incluir uma descrição de forma sucinta das estruturas de dados usadas no programa; descrição dos tipos abstratos de dados (TADs) definidos, explicitando as operações disponíveis; explicações de como os TADs são implementados; funcionamento das principais funções e procedimentos utilizados; o formato de entrada e saída dos dados, como executar o programa e as decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado;
3. descrição dos testes realizados;
4. estudo da complexidade: estudo da complexidade (notação O) do tempo de execução dos métodos implementados e do programa como um todo;
5. conclusão: avaliação do grupo sobre o trabalho considerando a experiência adquirida, a contribuição para o aprendizado da disciplina, as principais dificuldades encontradas ao implementá-lo e como tais dificuldades foram superadas;
6. bibliografia: cite as fontes consultadas na resolução do trabalho.

Todos os artefatos (códigos fontes e relatório) devem ser entregues conforme as instruções contidas nesse documento.

Na seção 3 são descritos os critérios de correção desta atividade.

### 3. Critérios de Correção

Conforme descrito no plano de ensino, os critérios de avaliação do trabalho são:

1. somente serão corrigidos os trabalhos com códigos fontes portáteis e sem de erros de compilação;
2. análise código fonte: modularização, uso adequado de comentários, legibilidade e indentação do código; (10%)
3. uso adequado de TAD e estruturas de dados; e (20%)
4. execução correta numa bateria de testes práticos; (50%)
5. documentação. (20%)
6. A critério do professor poderá ser agendada uma apresentação oral do trabalho. Na ausência de plágio, as notas dos trabalhos serão computadas da seguinte forma:  $\text{nota\_apresentação} * \text{nota\_trabalho}$ , ou seja, a nota final é ponderada pela nota da apresentação.

Na seção 4 indica-se a bibliografia consultada para a confecção deste documento.



Instituto Federal de Educação, Ciência e Tecnologia  
Curso: Bacharelado em Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Professor: Mário Luiz Rodrigues Oliveira  
Atividade: 2ª Trabalho Prático  
Formiga, MG, 24 de outubro de 2016

#### **4. Bibliografia**

Goodrich, Michael T.; Tamassia, Roberto. Estruturas de Dados & Algoritmos. 5ª edição. Porto Alegre: Bookman, 2013.

Preiss, Bruno R. Estruturas de Dados e Algoritmos - Padrões de projetos orientados a objetos com Java. Rio de Janeiro: Campus, 2000.