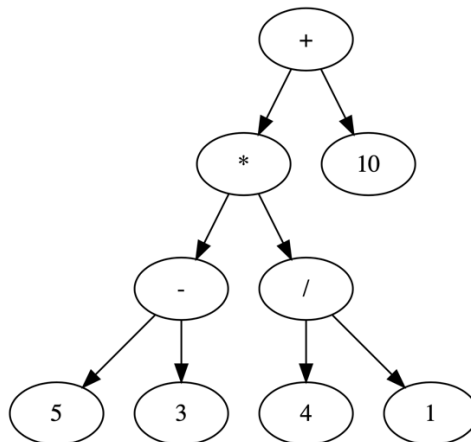


Universidade Federal do Espírito Santo – Departamento de Informática
Estruturas de Dados I (INF09292)
2º Trabalho Prático
Período: 2020/1 EARTE
Professora Patrícia Dockhorn Costa

Neste trabalho você implementará um programa para avaliar expressões aritméticas por meio de uma **Árvore Binária de Expressões Aritméticas**.

O que é uma árvore binária de expressões aritméticas?

Uma árvore binária de expressões aritméticas é uma estrutura de dados que permite avaliar (calcular) o valor da expressão por meio de sua organização hierárquica. Nesta árvore, nós folha são operandos (números) e nós internos representam os operadores (*, /, +, -). Neste trabalho, suportaremos apenas operadores binários, ou seja, que utilizam 2 números (ou duas sub-expressões, também binárias). Considere a expressão aritmética $((5-3)*(4/1))+10$. A árvore de expressões equivalente a esta expressão é a seguinte:



Como montar a árvore a partir de uma expressão em formato de string?

O processo de interpretar uma *string* contendo uma expressão e a transformar em uma árvore binária é denominado de “*parsing*”. Para simplificar a lógica do algoritmo do *parsing* dessas expressões, faremos uso extensivo de parênteses (por vezes, desnecessários):

- Todo número é cercado por parênteses;
- Todo operador e seus dois números serão cercados por parênteses;
- Todo operador e suas duas sub-expressões serão cercados por parênteses.

Portanto, a expressão anterior estaria no formato: $(((((5)-(3)))*((4)/(1)))+(10))$.

O seu programa deve:

- ler as expressões de um arquivo texto de entrada (em formato de string);
- fazer o parsing das strings e montar as árvores binárias de expressões na memória;
- calcular os valores das expressões por meio das árvores; e,
- escrever os valores das expressões em um arquivo de saída.

Entrada do Programa:

Um arquivo texto contendo as expressões, uma por linha, no formato indicado acima. Para simplificar, assuma que não há erros de sintaxe nas expressões. Exemplo de arquivo de entrada:

```
(( ( (5) - (3) ) * ( (4) / (1) ) ) + (10) )  
(( ( (10) * (3) ) + (5) ) / ( (10) - (5) ) )  
(( ( (47) - (7) ) * (2) ) - (3) )  
((( (120) * (2) ) - (100) ) * (10) )
```

Saída do Programa:

O valor calculado das expressões, uma por linha, seguindo a ordem do arquivo de entrada. Exemplo de arquivo de saída:

```
18  
7  
77  
1400
```

Arquivos Esperados pelo *Classroom*:

- 1) TAD Árvore de Expressões (arv.h e arv.c)
- 2) Arquivo do cliente <seunome.c>

Bom Trabalho!!!!