
TRABALHO PRÁTICO 1

Construa um tradutor que leia um arquivo contendo expressões aritméticas em notação infixada, descritas de acordo com a gramática abaixo, e as traduza para uma notação pós-fixada, conforme o exemplo fornecido.

```
program → expression block
block   → { decls stmts }
decls   → decls decl
        | ε
decl    → type id;
stmts   → stmts stmt
        | ε
stmt    → block
        | expr;
expr    → expr + term
        | expr - term
        | term
term    → term * factor
        | term / factor
        | factor
factor  → (expr)
        | num
        | id
```

Exemplo de tradução:

```
// arquivo fonte
expression
{
    int x;
    int val;
    x + val;
    2+(3-5);
    {
        float x;  /* outro x */
        2 - 3.5 + x;
    }
    x;
}
```

```
(x:int)(val:int)+
(2)(3)(5)-+
(2)(3.5)-(x:float)+
(x:int)
```

INSTRUÇÕES

O tradutor deve ser composto por um **analisador léxico**, um **analisador sintático preditivo** construído a partir de um esquema de tradução dirigido por sintaxe, e utilizar uma **tabela de símbolos por escopo**, como visto na disciplina.

O analisador léxico deve ignorar:

- Espaços, tabulações e saltos de linha
- Comentários iniciando com `//` e indo até o final da linha
- Comentários iniciando com `/*` e finalizando com `*/`

O analisador sintático deve aceitar:

- Números inteiros e pontos flutuantes para o terminal **num**
- Tipos `int` e `float` para o terminal **type**
- Nomes de variáveis compostos apenas por letras do alfabeto para o terminal **id**
- A palavra “expression” para o terminal **expression**

Erros devem ser reportados:

- Através do número da linha e uma descrição do erro
- Variáveis usadas sem declaração devem gerar erro

O tradutor deve ler um arquivo texto com a extensão “.exp”, passado na linha de comando, e exibir o resultado da tradução na tela. Ao receber a opção “--help” ele deve exibir instruções de como usá-lo, seguindo o padrão de outros programas do sistema operacional Linux.

```
> ./tradutor teste.exp
(x:int)(val:int)+
(2)(3)(5)-+
(2)(3.5)-(x:float)+
(x:int)
```

ENTREGA DO TRABALHO

Trabalho: individual

Data da entrega: 26/11/2018 (até a meia noite)

Valor: 3,0 pontos (na 1a Unidade)

Forma de entrega: enviar os arquivos fonte (.cpp) e os arquivos de inclusão (.h) compactados no formato **tar.gz** através da tarefa correspondente no SIGAA.

O não cumprimento das orientações resultará em **penalidades**:

- Programa não compila (g++ 7.3.0 ou superior) ou executa (Linux) (3,0 pontos)
- Programa contém partes de outros trabalhos (3,0 pontos)
- Atraso na entrega (1,5 pontos por dia de atraso)
- Arquivo compactado em formato diferente do solicitado (0,5 ponto)
- Envio de outros arquivos que não sejam os de código (0,5 ponto)
- Programa sem comentários e/ou desorganizado (0,5 ponto)