

# Projeto de Compiladores

## Prof. Dr. Francisco Isidro

### Instruções para Entrega do Trabalho:

- Este trabalho deverá ser desenvolvido em grupos de, **no máximo**, 3 alunos.

### Enunciado:

Cada grupo deve definir a sua própria gramática (baseada no exemplo a seguir) e os tokens necessários. Os requisitos **mínimos** são:

- Deve ter pelo menos 2 tipos de variáveis.
- Deve ter a estrutura de controle *if ... else*.
- Deve ter a estrutura de repetição *while* e *do ... while*.
- A parte de expressões envolvendo os operadores matemáticos deve ser realizada de maneira correta, respeitando a precedência.
- As atribuições também devem ser realizadas. Não é necessário verificar se é possível realizar as operações, devido aos tipos das variáveis.
- Os comandos de leitura do teclado e de impressão na tela devem ser disponibilizados.
- O compilador tem que aceitar números decimais.
- A cada utilização de uma variável, é necessário verificar se a mesma já foi declarada.

O compilador deve fazer a conversão de um programa desenvolvido na Linguagem definida pelo grupo para a Linguagem C ou Java (fica a critério do grupo definir para qual linguagem).

A verificação da corretude do programa será realizada compilando o arquivo gerado pelo compilador desenvolvido.

Seu compilador deverá receber como entrada um arquivo contendo um programa escrito na Linguagem definida pelo grupo e gerar uma forma equivalente em C ou Java, que deverá ser compilada em um compilador qualquer (DevC++, CodeBlocks, MS-Visual C, Gcc, javac) e não deverá conter erros.

OBS: a gramática não pode conter recursividade à esquerda e produções vazias (que porventura venham a surgir). Caso seja necessário, efetue sua fatoração à esquerda.

### Exemplo de um Compilador

A descrição a seguir ilustra um exemplo de um Compilador que faz a conversão de um programa desenvolvido em uma linguagem fictícia para uma forma equivalente na linguagem C.

Os termos em **negrito** significam palavras reservadas. Preste muita atenção aos sinais de pontuação. Considere todas as variáveis como sendo do tipo INTEIRO (inicialmente).

Prog → **programa** Declara Bloco **fimprog.**  
 Declara → **declare** Id (, Id)\* .  
 Bloco → (Cmd. )+  
 Cmd → CmdLeitura | CmdEscrita | CmdExpr | CmdIf  
 CmdLeitura → **leia**( Id )  
 CmdEscrita → **escreva**( Texto | Id )  
 CmdIf → "**se**" '(' Expr Op\_rel Expr ')' **entao** '{' Cmd+ '}' ("**senao**" '{' Cmd+ '}' )?  
 CmdExpr → Id := Expr  
 Op\_rel → '<' | '>' | "<=" | ">=" | "!=" | "=="  
 Expr → Expr + Termo | Expr - Termo | Termo  
 Termo → Termo \* Fator | Termo / Fator | Fator  
 Fator → Num | Id | ( Expr )  
 Texto → "(0..9 | a..z | A..Z | ' ' | )+ "  
 Num → (0..9)+  
 Id → (a..z | A..Z) (a..z | A..Z | 0..9)\*

OBS: espaços em branco, tabs e enter podem aparecer e devem ser eliminados.

Entrada:input.in

```

programa
declare a,b,c,d.
escreva ("Programa Teste").
escreva ("Digite A").
leia (a).
escreva ("Digite B").
leia (b).

se (a<b)
{
  c := a + b.
}senao
{
  c := a - b.
}

escreva ("C e igual a ").
escreva (c).

d := c * a + b.

escreva ("D e igual a ").
escreva (d).

fimprog.
  
```



Saída:input.c

```

#include <stdio.h>

void main(void)
{
  int a,b,c,d;
  printf("Programa Teste");
  printf("Digite A");
  scanf("%d",&a);
  printf("Digite B");
  scanf("%d",&b);

  if (a<b)
  {
    c = a + b;
  }else
  {
    c = a - b;
  }

  printf("C e igual a ");
  printf("%d",c);

  d = c * a + b;

  printf("D e igual a ");
  printf("%d",d);

}
  
```