CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Departamento de Computação

Disciplina	**Curso**	**Turno**	**Período**	**Valor**
Lab. Linguagens de Programação	Engenharia da Computação	Tarde	4º	15 Pontos

Aluno(a)		**Matrícula**	
Aluno(a)		**Matrícula**	

Ao entregar esse exame eu dou minha palavra que eu o fiz somente com minha dupla, entendendo que eu posso consultar qualquer material publicamente disponível, exceto aqueles disponibilizados por colegas que estão fazendo esse curso ou que já o fizeram no passado.

Prova 1

1ª Questão) (15pts) A linguagem de programação C possui tipos de dados **simples**, como `char`/`short`/`int`/`long`/`float`/`double`, e **compostos**, como `struct`/`union`. Esses dois tipos compostos permitem a declaração de registros, ou seja, estruturas compostas por um ou mais tipos de dados. Essas estruturas permitem declarar quaisquer tipos de dados como membros, inclusive outras `struct`'s e `union`'s – ou seja, permite declarações aninhadas. Considere o seguinte trecho de código com diversas declaração de variáveis, incluindo tipos simples e compostos combinados:

```
01: struct A {
02:     char b;
03:     union C {
04:         struct {
05:             short d;
06:             int e;
07:         } f;
08:         long g, h;
09:         union {
10:             float i;
11:             double j;
12:         } k, l;
13:     } c;
14: } a;
```

```
15: long m, n, o;
16: char p;
17: union {
18:     struct T {
19:         int u;
20:         char v;
21:     } t;
22:     int w;
23:     union {
24:         long x;
25:     } y;
26: } z;
27: float q, r;
28: double s;
```

Considere as seguintes características das declarações:

- Toda declaração de variável possui um tipo (dentre os listados acima), define um ou mais nomes e termina com ponto-vírgula. Por exemplo:
  - `char p;`
  - `long m, n, o;`
  - `union { ... } k, l;`
- As estruturas (`struct`) e uniões (`union`) podem ter nome ou não (anônimas), mas sempre devem definir pelo menos um nome de variável depois de fecha chaves. Por exemplo:
  - `union C { ... } c;`
  - `struct { ... } f;`

Dica: Não é preciso garantir a unicidade dos nomes das variáveis, ou seja, que nomes iguais não entram em conflito com outros nomes. Contudo, será dado até 2pts extras caso as declarações sejam código C válido.

a) (5pts) Desenhe um autômato para separar os elementos léxicos dessa linguagem.
b) (5pts) Descreva uma gramática (notação *EBNF*) para fazer o *parsing* dessa linguagem.
c) (5pts) Desenvolva um programa capaz de identificar se um arquivo contendo declarações de variáveis está bem formado, respondendo **Sim** em caso positivo e **Não** caso contrário. O programa deve receber o arquivo por parâmetro em linha de comando. O *parser* deve ser desenvolvido de forma própria, não sendo permitido o uso de bibliotecas de terceiros para esse propósito.