



Compiladores: Prova 1

Nome:

Matrícula:

Data:

Observações:

- (a) A prova é individual e sem consulta, sendo vedado o uso de calculadoras e de telefones celulares.
- (b) A interpretação dos comandos das questões faz parte da avaliação.
- (c) A nota da prova será igual a 0 (zero) caso o estudante consulte algum material durante a prova, ou receba ou ofereça qualquer ajuda a outro estudante durante a prova.
- (d) As respostas podem ser marcadas à lápis. Em uma questão de múltiplas escolhas, múltiplas marcações anularão a referida questão.
- (e) O grampo da prova não deve ser removido. Caso o grampo seja removido, a nota da prova será igual a 0 (zero).

Parte A. (70 pontos) Assinale a alternativa correta.

4. Considere a gramática livre de contexto G dada abaixo:

$$B \rightarrow A x \mid y B \mid z$$

$$A \rightarrow B z$$

1. Cada parte da compilação contém quantas fases?

- (A) 2
- (X) 3
- (C) 4
- (D) 6

Quantos são os não-terminais de G ?

- (A) 1
- (X) 2
- (C) 3
- (D) 4

2. Qual fase da compilação é a responsável pela geração da árvore gramatical?

- (A) análise léxica
- (X) análise sintática
- (C) análise semântica
- (D) geração de código

5. Sejam \odot e \oslash dois operadores binários tais que $x \odot y = x \times (y + 1)$ e $x \oslash y = y - x$. Determine o valor da expressão

$$2 \odot 3 \odot 5 \oslash 7 \oslash 9$$

3. Assumindo que a multiplicação tem a mesma precedência da divisão, qual é a forma posfixa da expressão infixa $7 \times 3 \div 5$?

- (A) $53 \times 7 \div$
- (B) $53 \div 7 \times$
- (X) $73 \times 5 \div$
- (D) $73 \div 5 \times$

assumindo que ambos operadores são associativos à direita e que \odot tem maior precedência em relação à \oslash .

- (A) 50
- (B) 64
- (C) -10
- (X) -36

6. Considere a gramática livre de contexto G abaixo:

$$\begin{aligned} E &\rightarrow T \rho E \mid T \delta E \mid T \\ T &\rightarrow T \psi F \mid T \phi F \mid F \\ F &\rightarrow \text{num} \mid \text{var} \end{aligned}$$

É correto afirmar que

- (A) δ tem maior precedência em relação a ψ
- (X) ϕ tem maior precedência em relação a ρ
- (C) ψ é associativa à direita
- (D) ρ é associativa à esquerda

7. Considere as duas afirmações abaixo.

- I. Analisadores sintáticos top-down constroem a árvore gramatical partindo das folhas em direção à raiz.
- II. Analisadores sintáticos bottom-up constroem a árvore gramatical partindo da raiz em direção às folhas.

Podemos afirmar que

- (A) ambas afirmações estão corretas
- (X) ambas afirmações estão incorretas
- (C) apenas a afirmação I está correta
- (D) apenas a afirmação II está correta

8. Considere a gramática livre de contexto abaixo.

$$\begin{aligned} S &\rightarrow aT \mid bSa \\ T &\rightarrow a \mid bT \mid cSa \end{aligned}$$

Qual é o número de nós da árvore sintática associada a cadeia $abba$?

- (A) 6
- (B) 7
- (X) 8
- (D) 9

9. Considere as duas afirmações abaixo, relativas à análise gramatical descendente recursiva.

- I. Cada não-terminal da gramática é associado a um procedimento.
- II. A sequência de chamada de procedimentos no processamento da entrada determina, de forma explícita, a árvore gramatical.

Podemos afirmar que

- (A) ambas afirmações estão corretas
- (B) ambas afirmações estão incorretas
- (X) apenas a afirmação I está correta
- (D) apenas a afirmação II está correta

10. Considere as gramáticas G_1 e G_2 , dadas por

$$A \rightarrow Aa \mid Ab \mid a$$

e

$$B \rightarrow Ba \mid Bb \mid b$$

respectivamente. Em relação às produções de ambas gramáticas, podemos afirmar que:

- (A) ambas gramáticas possuem produções recursivas à esquerda
- (X) ambas gramáticas possuem produções recursivas à direita
- (C) G_1 possuem produções recursivas à esquerda, G_2 possui produções recursivas à direita
- (D) G_1 possuem produções recursivas à direita, G_2 possui produções recursivas à esquerda

11. Considere as afirmações abaixo.

- I. Em uma dada gramática, uma sequência de tokens é denominada lexema.
- II. A sequência de caracteres que compõem um único token é denominada sentença.

Podemos afirmar que

- (A) ambas afirmações estão corretas
- (X) ambas afirmações estão incorretas
- (C) a afirmativa I está correta e a afirmativa II está incorreta
- (D) a afirmativa I está incorreta e a afirmativa II está correta

12. Marque a opção abaixo que liste todos os valores- L da expressão $x = z + y$.

- (X) x
- (B) y
- (C) x e z
- (D) y e z

13. Considere o código de uma expressão para uma máquina de pilha abstrata, onde $a \text{ DIV } b$ computa o quociente da divisão inteira de a por b e $a \text{ MOD } b$ computa o resto da divisão inteira de a por b :

```
push 7
push 3
push 5
MOD
DIV
```

De acordo com as convenções adotadas, qual seria o resultado da execução deste código pela máquina de pilha abstrata?

- (A) 0
- (B) 1
- (X) 2
- (D) 3

14. Considere a construção parcial do gabarito de tradução do comando `do-while` da linguagem C:

`do-while`

<code>rotulo block</code>
código para <code>block</code>
código para <code>expr</code>
<code>??? block</code>

Qual instrução de fluxo da máquina de pilha deve substituir a marcação `???` para completar corretamente o gabarito acima?

- (A) `goto`
- (B) `rotulo`
- (X) `gotrue`
- (D) `gofalse`

Parte B. (30 pontos) Resolva as questões a seguir.

15. Um número inteiro não-negativo n pode ser representado em base 5 por meio de k dígitos d_i , onde

$$n = d_0 + d_1 \times 5 + d_2 \times 5^2 + \dots + d_{k-1} \times 5^{k-1},$$

onde $d_i \in \{0, 1, 2, 3, 4\}$ para $i = 0, 1, \dots, k-1$.

Implemente, em C, C++ ou Python, uma função chamada `scanner` que receba como parâmetro uma string `s` e que retorne o valor, em base decimal, do número em base 5 da maior sequência de dígitos em base 5, consecutivos, contidos em `s`, a partir de seu primeiro caractere. Caso esta sequência tenha tamanho igual a zero, retorne o valor zero.

Seguem abaixo alguns testes unitários para esta função.

```
assert(scanner("123") == 38);
assert(scanner("004x") == 4);
assert(scanner("abc") == 0);
assert(scanner("12345") == 194);
assert(scanner("n4321") == 0);
assert(scanner("") == 0);
```

Importante: Escreva o código com letra legível, de forma organizada e clara, numerando as linhas. O código não deve exceder 40 linhas.

```
1 // Solução em C
2 int scanner(const char *s)
3 {
4     int valor = 0, i, N = strlen(s);
5
6     for (i = 0; i < N; ++i)
7     {
8         // Se encontrar um caractere inválido finaliza a identificação do lexema
9         if (s[i] < '0' || s[i] > '4')
10             break;
11
12         // Atualiza o valor
```

```
13     valor = valor*5 + (s[i] - '0');
14 }
15
16 return valor;
17 }
```

```
1 # Solução em Python
2 def scanner(s):
3
4     from itertools import takewhile
5
6     # Extraí o maior prefixo com os dígitos válidos
7     # O zero à esquerda trata do caso do prefixo vazio
8     lexema = '0' + ''.join(takewhile(lambda c : c in "01234", s))
9
10    # Converte o prefixo para base 5
11    return int(lexema, 5)
```