

Universidade Federal do Rio Grande do Norte
 Unidade Acadêmica Especializada em Ciências Agrárias
 Análise e Desenvolvimento de Sistemas
 Componente Curricular: Fundamentos de Computação
 Prof. Josenalde Oliveira
Lista de Exercícios 6

1. Verifique se os trechos de código abaixo possuem algum tipo de erro e, se existir, identifique-os, classificando em **erro sintático** ou **erro semântico** ou **ambos**. Lembre que erros semânticos também estão associados a problemas de lógica apenas identificados durante a execução, ou seja, o código compila normalmente mas não cumpre o que se espera ou se comporta de maneira indesejada/estranha. Erros semânticos também pode ser associados à má interpretação pelo compilador.

a)	Análise:
<pre>int 1_x; int y[] = {9,10}, i; for (i=0; i<=2; i++) cout << y[i];</pre>	
b)	Análise:
<pre>short x = 5, y = 10; switch() { case 1: cout << "ok"; break; case 0: cout << "not ok"; break; }</pre>	
c)	Análise:
<pre>unsigned char i = 0; while (i <= 255) { cout << i << endl; i++; Sleep(1); } cout << "FIM\n";</pre>	
d)	Análise:
<pre>char str[] = {'t','a','d','s'}; puts(str); putchar('\n');</pre>	
e)	Análise:
<pre>int x=2, y=3; while (x+y) cout << "01";</pre>	

OBS: nos códigos acima, suponha que todos os #includes necessários estão definidos.

2. O analisador léxico (scanner) é a primeira parte do processo de compilação e consiste em percorrer as palavras (lexemas) do código fonte, separando-as e identificando os

respectivos tipos dos lexemas, com base numa tabela de símbolos. Neste processo são geradas tuplas <tipoLexema, Lexema>, que são denominados TOKENS. Estes tokens são então passados para a segunda etapa, o analisador sintático (parser). Com base neste conceito e nos assuntos discutidos em sala de aula, reescreva o código fonte que gerou a sequência de tokens indicada:

a)
<tipoDado, "int"> <id, "x"> <operador, "="> <num, "5"> <terminador, ";"> <espaço, "\n">
<id, "x"> <operador, "++"> <terminador, ";">

b)
<comando, "while"> <delimitador, "("> <num, "1"> <delimitador, ")"> <comando, "cout">
<operador, "<<"> <texto, "ufrn"> <terminador, ";">

3. Suponha o seguinte código fonte na linguagem C++. Escreva a saída do analisador léxico para este código, sabendo que os mesmos tipos da questão 2 constam na tabela de símbolos. Considere também os lexemas do tipo tabulação (\t) e nova linha (\n).

a)
int x=2,y=3;
if (x+y<10){
 x+=y;
 ++y;
}

b)
float double x;
x=1.20
(x>1.0?x++:x--;

4. Na letra b) da questão acima, o analisador sintático ao receber os tokens irá gerar algum erro? Se sim, quais? Execute este código e anote as mensagens informadas pelo compilador.

5. Marque V (Verdadeiro) ou F (Falso) nas declarações abaixo. Quando for Falso, justifique.

() O código fonte abaixo soma os números 10 e 17 e exibe o resultado na tela. Este é um exemplo de código em linguagem de primeira geração.

```
li $s0, 10
li $s1, 17
add $s2,$s1,$s0
la $a0, ($s2)
li $v0,1
syscall
li $v0, 10
syscall
```

() Conhecer os algoritmos de conversão DEC<->BIN e operações aritméticas no sistema binário são fundamentais para um programador de primeira geração.

() O SGBD Postgres, embora possua GUI para sua administração, pode ser administrado em linguagem de quinta geração.

() A instrução SELECT * FROM ALUNOS é um exemplo de comando de linguagem de quarta geração.

() Um programa na linguagem Java não necessariamente precisa estar definido dentro de uma classe, por ser linguagem naturalmente OO.

() O paradigma estruturado permite tornar o código mais modularizado que o paradigma OO e, assim, facilita o reuso de código fonte.

() As chamadas linguagens de script na WEB (Javascript, por exemplo) e linguagens e bibliotecas usadas no desenvolvimento de jogos (allegro C++, Unity C#, Python pygame, MIT app inventor etc.) baseiam-se no paradigma de orientação a eventos, por possuírem métodos associados às interações do usuário.

() O paradigma orientado a eventos não inclui ou está relacionado ao paradigma OO. Orientação a eventos é subconjunto de OO. Já OO não implica OE.

() O trecho de código abaixo é um exemplo típico de Polimorfismo no paradigma OO

```
class Caixa {
public:
    int altura, largura;
    void Altura(int a) { altura=a; }
    void Largura(int l) { largura=l; }
};

class CaixaColorida : public Caixa {
public:
    int cor;
    void Cor(int c) { cor=c; }
};

void main()
{
    CaixaColorida cc;
    cc.Cor(5);
    cc.Largura(3); // herdada
    cc.Altura(50); // herdada
}
```

6. Quais as 04 características fundamentais do paradigma OO? Descreva os principais aspectos de cada uma e apresente exemplos de implementação de cada característica em sua linguagem de programação de preferência (quando possível e existente na linguagem de interesse).