

PLC24-Exe3

5 Questions

1. Avalie a veracidade da seguinte afirmação:

O módulo lex do python permite reconhecer símbolos e tokenizar frases de uma linguagem.

29/29 ☒ True

0/29 ☐ False

2.

Avalie a veracidade da seguinte afirmação:

O módulo lex do python possui regras pré-definidas para controlar erros relativos a caracteres inesperados, mas não possui regras para expressões a ignorar.

6/30 ☐ True

24/30 ☒ False

3. Selecione as afirmações (1 ou mais) abaixo que são verdadeiras

23/29 ☒ **A** **LexToken(NUMBER,2,7,1)** pode ser um elemento do output produzido pela execução de um analisador léxico gerado com o lex e representa um símbolo.

22/29 ☒ **B** Considerando que o output do analisador léxico gerado pelo lex é:
LexToken(NUMBER,2,7,1)
pode concluir-se que NUMBER é o identificador, ou tipo, do símbolo reconhecido, cujo texto concreto é "2".

3/29 ☐ **C** Considerando que o output do analisador léxico gerado pelo lex é: **LexToken(NUMBER,2,1)**
pode concluir-se que "NUMBER" é a palavra do texto fonte identificada com o código 2.

17/29 ☒ **D** Considerando que o output do analisador léxico gerado pelo lex é: **LexToken(ID, 'NUMBER',2,7)**
pode concluir-se que foi reconhecido no input um símbolo denotado pela designação 'ID' e reconhecido ao ler a palavra 'NUMBER' na linha 7.

4. Considere o seguinte fragmento de um programa *python* que está corretamente escrito fazendo uso do gerador *lex* do módulo *ply*:

```
import ply.lex as lex
token = ('PLUS', ....., 'RPAREN')
t_PLUS = r'\+'
t_MINUS = r'\-'
t_TIMES = r'\*'
t_DIVIDE = r'\/'
t_LPAREN = r'\('
t_RPAREN = r'\)'

t_ignore = '\t'
def t_error(t):
    print("Erro '%s'" % t.value[0])

lexer = lex.lex()
linha = "...."

lexer.input(frase)
for tok in lexer:
    print(tok)
```

Selecione, abaixo, as afirmações verdadeiras .

- 22/30 **A** Se frase for "+ - ** /" o analisador léxico *lexer* analisa corretamente a frase e retorna 5 símbolos.
- 4/30 **B** Se frase for "+ - ** /" o analisador léxico *lexer* analisa corretamente a frase e retorna 4 símbolos.
- 7/30 **C** Se frase for "1+23 - 4*5*22/66" o analisador léxico *lexer* analisa corretamente a frase e retorna 5 símbolos.
- 19/30 **D** Se frase for "1+23 - 4*5*22/66" o analisador léxico *lexer* termina intempestivamente logo ao ler o primeiro carater.

5. Selecione as afirmações (1 ou mais) abaixo que são verdadeiras

- 28/30 **A** Caso use uma função para definir uma regra em *lex*, é possível alterar o token e retorná-lo
- 22/30 **B** Caso use uma função para definir uma regra em *lex*, é possível ignorar o token identificado por essa regra
- 0/30 **C** Na função de erro
def t_error(t):
 print("Erro '%s'" % t.value[0])
 t.lexer.skip(1)
a linha `t.lexer.skip(1)` é inútil e poderia ser removida sem alterar o comportamento do programa
- 23/30 **D** As regras definidas utilizando funções tem prioridade sobre as outras apenas definidas por string