Universidade Federal do Ceará – UFC

Lista 1 – Engenharia da Computação

Disciplina: Paradigmas de Linguagens de Programação **Professor:** Joniel Bastos

Nome:_____ **N:**_____

Parte 1 – Análise Léxica

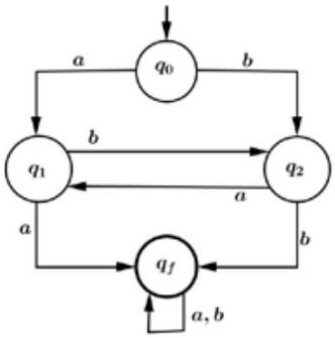
1. Escreva um código que reconheça os comentários em um código, considerando /*...*/ para informar os comentários no código.

2. Escreva um código que reconheça se o número informado é inteiro ou tipo flutuante. Considere a "," como meio utilizado para informar que o número é decimal.

3. Considere o seguinte autômato M = ({a,b},{q0, q1, q2, qf}, δ, q0, {qf}), tal que:
   Σ = {a, b}
   Q = {q0, q1, q2, qf} são os estados possíveis
   δ : (Q × Σ) → Q é a função de transição
   Q = {qf} é o estado final
   Diagrama de estados:
   δ(q0, a) = q1

δ	**a**	**b**
q_0	q_1	q_2
q_1	q_f	q_2
q_2	q_1	q_f
q_f	q_f	q_f

   E que o seu diagrama do autômato é representado por



   Escreva um código que verifique se uma cadeia de caractere é reconhecida por esse autômato.

Parte 2 – Nomes (Escopos)

1. Escreva uma função em C que inclua a seguinte sequência de sentenças:

```
x = 21;
int x;
x = 42;
```

   Rode o programa e explique os resultados. Reescreva o mesmo código em C++ e Java e compare os resultados.

2. Considere o seguinte esqueleto de programa em C:

```
void fun1(void);  /* protótipo */
void fun2(void);  /* protótipo */
void fun3(void);  /* protótipo */
void main() {
  int a, b, c;
  ...
 }
void fun1(void) {
  int b, c, d;
  ...
 }
void fun2(void) {
  int c, d, e;
  ...
 }
void fun3(void) {
  int d, e, f;
  ...
 }
```

Dada as seguintes sequências de chamadas e assumindo que o escopo dinâmico é usado, que variáveis são visíveis durante a execução da última função chamada? Inclua com cada variável visível o nome da função na qual ela foi definida.

a) **main** chama **fun1**; **fun1** chama **fun2**; **fun2** chama **fun3**.
b) **main** chama **fun1**; **fun1** chama **fun3**
c) **main** chama **fun2**; **fun2** chama **fun3**; **fun3** chama **fun1**.
d) **main** chama **fun3**; **fun3** chama **fun1**.
e) **main** chama **fun1**; **fun1** chama **fun3**; **fun3** chama **fun2**.
f) **main** chama **fun3**; **fun3** chama **fun2**; **fun2** chama **fun1**.

3. Considere as seguintes funções em Python

```python
# Função exemplo 1
def func1():
    def foo():
        L = []
        def bar():
            L.append(5)
        bar()
        return L
    foo()

# Função exemplo 2
def func2():
    def foo():
        def bar():
            L.append(5)
        L = []
        bar()
        return L
    foo()
```

```python
# Função exemplo 3
def func3():
    def foo():
        L = []
        bar()
        return L
    def bar():
        L.append(5)
    foo()
```

Crie uma função para realizar a chamada das três funções e informar “Sucesso” se a execução ocorreu sem erro e “Falha” se houve erro. Em caso de erro, relate o porquê do erro.

OBS: A resolução deve ser enviada em um único arquivo zipado com o nome *LISTA_01_PARADIGMAS_FULANO_DE_TAL* pelo *classroom*.