

Observações

- O trabalho deve ser feito em grupos de no máximo 2 componentes
- Trabalhos entregues após a data limite não serão aceitos
- Data limite de entrega: **23 de Novembro de 2018 : 23h59m**
- Enviar o trabalho para o email: **erick.maziero@dcc.ufla.br**
 - O código e documentação devem estar em um **arquivo .zip**

Trabalho Prático

Nesse trabalho, o grupo deve implementar um programa que simule uma máquina de Turing determinística padrão que faça a computação de funções numéricas.

O programa deve ser implementado em Python (2.7 ou 3.*) ou Java. De tal forma que sua chamada se dê por linha de comando, seguindo o padrão a seguir:

Python

```
Python programa.py < entrada.txt
```

Java

```
java -jar programa.jar < entrada.txt
```

Na definição acima, `entrada.txt` é um arquivo texto, codificado em UTF-8, que contém a descrição de uma máquina de Turing e uma entrada a ser simulada na máquina de Turing descrita.

O arquivo `entrada.txt` formalizará uma máquina de Turing como uma quintupla $M = (Q, \Sigma, \Gamma, \delta, q_0)$:

- Q = conjunto de estados (padrão $q[0 - 9] +$)
- Σ = alfabeto de entrada ($\{1\}$ para representação unária dos argumentos numéricos)
- Γ = alfabeto da fita ($\{1, B\}$ o símbolo branco fará a separação entre os argumentos numéricos na fita)
- δ = função de transição no formato $(q_i, x) \rightarrow (q_j, y, D)$; assim, estando no estado q_i , lendo x , vai para o estado q_j , escreve y e movimenta na direção de D . D será L para esquerda ou R para direita.
- q_0 = estado inicial

Exemplo de uma descrição de M :

```
(
    {q0, q1, q2, q3, q4, q5},
    {1},
    {1,B},
    {
        (q0, B) -> (q1, B, R),
        (q1, 1) -> (q1, 1, R),
        (q1, B) -> (q2, 1, R),
        (q2, 1) -> (q2, 1, R),
        (q2, B) -> (q3, B, L),
        (q3, 1) -> (q4, B, L),
        (q4, 1) -> (q5, B, L),
        (q5, 1) -> (q5, 1, L)
    }
    {q0}
)
B1111B11B
```

Considerando o exemplo acima, com entrada *B1111B11B*, o programa deve apresentar a configuração da fita a cada transição de M , conforme a seguir:

```
{q0}B1111B11B
B{q1}1111B11B
B1{q1}111B11B
B11{q1}11B11B
B111{q1}1B11B
B1111{q1}B11B
B11111{q2}11B
B111111{q2}1B
B1111111{q2}B
B1111111{q3}1B
B11111{q4}1BB
B1111{q5}1BBB
B111{q5}11BBB
B11{q5}111BBB
B1{q5}1111BBB
B{q5}11111BBB
{q5}B11111BBB
```

A pontuação máxima do trabalho será de **10 pontos**, distribuídos nos itens a seguir:

- **3 pontos** para qualidade de código (legibilidade, modularização e documentação)
- **7 pontos** para percentual de acertos nos casos de teste