MAC0316/5754
Exercício Programa 1
Data de entrega: 02/11/2020, 8:00hs

Instruções:

1. Você deve entregar seu programa pelo e-Disciplinas em um **único arquivo** *.rkt* contendo as definições do interpretador
2. Programas **atrasados** terão uma **penalidade de 20%** na pontuação **POR DIA**.

Interpretador (10 pontos)

a) **(4 pontos)** Implemente `let`, `let*` e `letrec` como visto em classe. Suporte um número fixo de argumentos (1, 2 e 1, respectivamente).

b) **(2 pontos)** Implemente a primitiva `quote`, que gera um símbolo, com `'` no seu interpretador. Assim:

```
(interpS '(quote alan))
```

deve retornar

```
- symbol
(symV 'alan)
```

c) **(4 pontos)** Implemente a primitiva `load` que recebe um símbolo com o nome de um arquivo, abre o arquivo e interpreta todas as expressões simbólicas nele. Assim:

```
(interpS '(load (quote test.txt)))
```

deve avaliar todas as expressões simbólicas contidas em *test.txt*. Note que este arquivo irá conter "expressões normais", não chamadas ao interpretador ou ao parser. Isso significa que você mesmo deve implementar essas chamadas (como ilustrado no exemplo acima).

IMPORTANTE: Não será dado **crédito parcial** aos intens acima. Apenas aqueles que funcionarem receberão pontos (e.g. se você implementar `let` no parser mas não no interpretador, a implementação do `let` está incompleta).

IMPORTANTE: Você deve usar a versão `plai-typed` (**não** `plai`) do interpretador com fechamentos *closureTyped.rkt* (disponibilizado com este enunciado). Assim, você poderá utilizar os comandos de gerenciamento de arquivos do Scheme. (**Dica:** procure a documentação do comando `read`).