



## MATA56 Paradigmas de Linguagens de Programação Prova 2

Aluno:

1. Você tem 110 minutos para fazer esta prova.
2. Você pode consultar qualquer material que estiver com você, mas não pode usar computadores, qualquer outro equipamento eletrônico ou fonte externa.
3. Você só pode usar funções pré-definidas quando estas forem explicitamente mencionadas nos enunciados das questões, qualquer outra função auxiliar deve ser codificada por você.
4. Você pode usar cláusulas pedidas em outras questões para resolver uma dada questão. Por exemplo, você pode usar as cláusulas pedidas em Q1 para solucionar a questão Q2, mesmo sem ter resolvido a Q1.
5. Utilize caneta. Assine a prova e as folhas de resposta logo no início do exame (i.e., AGORA mesmo).
6. Coloque a caneta de lado assim que o professor anunciar o final da prova.
7. Qualquer violação das regras acima implica em nota zero na prova.

**Nenhuma das funções escritas nesta prova deve assumir o uso de registros de memórias modificáveis após a sua inicialização. Iterações devem ser realizadas somente com chamadas recursivas.**

### Questão 1 (2,0 pontos)

Escreva em Racket a função (**contagem** *N*) que recebe um número *N* e produz uma lista com contagem de um até deste número. Por exemplo:

```
> (contagem 10)
(1 2 3 4 5 6 7 8 9 10)
```

A função deve voltar uma lista vazia se *N* for menor que um.

### Questão 2 (2,0 pontos)

Escreva em Racket a função (**aplanar** *lg*) que recebe uma lista genérica *LG* e retorna uma lista simples *L* com todos elementos de *LG*. Por exemplo:

```
> (aplanar '(a (b c (a)) ((a)) (d a)))
(a b c a a d a)
```

**Questão 3 (2 pontos)** Escreva em Racket a função (**iguais11** *generic-list generic-list*) que recebe duas listas genéricas e retorna verdadeiro se e somente se elas forem exatamente iguais.

```
> (iguais-11 '((a b c) d (e (f) (g h (i)))) '((a b c) d (e (f) (g h (i))))
#t
```

### Questão 4 (2,0 pontos)

Escreva em Racket a função (**transforma** *f lg*) que recebe função *F* (*X*) e uma lista genérica *LG*, e aplica a função a todos os elementos *X* da *LG*. Por exemplo:

```
> (transforma (lambda(x) (* x x)) '(1 (2 3 (4)) ((5)) (6 7)))
'(1 (4 9 (16)) ((25)) (36 49)))
```

### Questão 5 (2,0 pontos)

Escreva em Racket a função (**remove-similares** *list [function]*) que remove os elementos similares entre si na lista. A função que determina similaridade é fornecida opcionalmente. Por default elementos são removidos se forem iguais. Por exemplo:

```
> (remove-similares '(a b d e d f r f a r a))
'(b e)
```