

Lista de Exercícios 4 – INF05008

Use os nomes de tipos de dados e funções definidos nas questões.

Data limite para submissão: 21 de setembro às 20:00

1. (Fácil) Em um baralho de UNO há dois tipos de cartas: cartas com números (de 0 a 9) e cartas especiais (que podem ser "Compra4", "Compra2", "Inverte", "PulaVez" e "TrocaCor"). Construa definições de dados para representar cartas de um baralho de UNO, completando as definições de dados a seguir, e defina 4 exemplos para cada tipo.

```
;; -----
;; TIPO CARTA-NUMERO:
;; -----
(define-struct carta-numero (cor valor))
;; Um elemento do conjunto Carta-numero é
;; ..... onde
;; ... : String, é a cor da carta, que pode ser "azul", "verde", "amarelo" ou "vermelho"
;; ... : Número, ....

;; -----
;; TIPO CARTA-ESPECIAL:
;; -----
(define-struct carta-especial (cor tipo))
;; Um elemento do conjunto Carta-especial é
;; ..... onde
;; ... : String, é a cor da carta, que pode ser "azul", "verde", "amarelo", "vermelho" ou "qualquer"
;; ... : String, ....

;; -----
;; TIPO CARTA:
;; -----
;; Um elemento do conjunto Carta é
;; 1. ....;
;; 2. ....
```

2. (Média) Construa a função `carta-cor` que, dada uma carta, devolve sua cor, em inglês. Ou seja, as cartas das cores "azul", "verde", "amarelo" ou "vermelho" devem retornar as cores "blue", "green", "yellow" e "red", respectivamente. Se a carta for da cor "qualquer", devolve "black". *Dica: Ao invés de construir uma função enorme, pense em decompor o problema...*
3. (Difícil) Desenvolva a função `jogada-valida` que, recebe uma carta, representando a carta da mesa, e uma outra carta ou um string, representando a carta da mão ou o fato de não haver carta na mão (no caso de ser um string), nesta ordem, e verifica se é possível jogar a carta da mão de acordo com as regras do jogo Uno (pesquise as regras de UNO). Se a carta da mão for um string, a função deve devolver falso. Assuma que, se a carta da mesa for uma carta especial colorida, uma carta de qualquer cor pode ser jogada. Se a carta for especial de alguma cor, deve ser jogada uma carta desta cor (ou do mesmo tipo). O resultado deve ser um valor booleano.
4. (Fácil) Defina o tipo de dado `Mão`, que pode ter até 5 cartas. Cada posição da mão pode conter uma carta ou estar livre (neste caso, a posição deve conter a palavra "livre"). Defina 2 constantes do tipo `Mão`.
5. (Fácil) Construa a função `define-jogada` que, dada uma mão e uma carta da mesa, nesta ordem, devolve uma carta da mão que pode ser jogada sobre esta carta da mesa. Se não houver nenhuma carta que pode ser jogada, a função deve devolver a mensagem "Jogada impossível". Para realizar a seleção, deve-se considerar a ordem das posições na definição da mão: cada carta tem prioridade sobre cartas em posições subsequentes.
6. (Média) Faça a função `desenha-carta` que, dada uma carta, gera uma imagem para esta carta, inspirada nas cartas de um jogo de UNO. *Obs: Não é para copiar uma imagem de uma carta da internet, a ideia é você desenhar a carta usando funções do pacote de imagens. Ver <https://docs.racket-lang.org/teachpack/2htdpimage.html>*
7. (Desafio - Ponto extra) Desenvolva a função `mostra-jogada` que, dada uma mão e uma carta, representando a carta da mesa, gera uma imagem mostrando a mão, a carta da mesa e a carta selecionada para jogar, caso haja uma carta para jogar. Se não for possível jogar nenhuma carta da mão, a saída deve ser a imagem da mão, da carta da mesa e a mensagem "Jogada impossível". *Dica: Decomponha o problema em problemas menores e construa a solução através da composição das soluções dos problemas menores.*