

### Exercícios

**Obs.:** Você não deve utilizar recursos do Racket que não estão disponíveis na versão *Beginning Student* da linguagem.

**Exercício 1.** Cada produto de um supermercado é definido por seu nome, um código de barras, seu preço e um possível desconto. Defina uma estrutura chamada **produto** para modelar isso. Após definir a estrutura, defina pelo menos 10 constantes do tipo produto. Os nomes e tipos esperados para os atributos devem ser:

- nome: String, representa o nome do produto
- cod: Número, código identificador do produto, representado por 4 dígitos
- preco: Número, preço do produto em reais
- desconto: Número, porcentagem de desconto a ser aplicado, que varia de 0 a 100

**Obs.:** Os códigos dos produtos devem ser únicos, então não crie 2 produtos com o mesmo código.

**Exercício 2.** Defina um tipo de dados chamado **inventario** para armazenar as informações de todos os produtos que um mercado possui. Um inventario deve conter um número arbitrário de produtos, e não deve conter produtos repetidos. Defina 2 constantes do tipo Inventário, sendo que uma deve ter pelo menos 10 produtos.

**Exercício 3.** Crie uma função chamada **nome\_produto** que, dado um inventario e um código de barras, retorna o nome do produto cadastrado no inventário com o código dado. Caso não haja um produto cadastrado com este código, retorna a string `“Produto não encontrado”`.

**Exercício 4.** Crie uma função chamada **verifica\_preco** que, dado um inventario e um código de barras, retorna o preço do produto cadastrado no inventário com o código dado, com o desconto aplicado. Caso não haja um produto cadastrado com este código, retorna a string `“Produto não encontrado”`.

**Exercício 5.** Defina um tipo de dados chamado **carrinho** para armazenar informações de um carrinho de compras de um cliente. Um carrinho deve conter um número arbitrário de códigos de barras.

**Obs.:** Um carrinho pode conter códigos repetidos, caso o cliente tenha comprado vários itens de um mesmo produto.

**Exercício 6.** Crie uma função chamada `checkout` que, dado um carrinho de compras e um inventário, retorna o valor total que o cliente deve pagar pelos produtos contidos no carrinho de acordo com o inventário (com os descontos aplicados). Para facilitar, assuma que todos os códigos contidos no carrinho constam no inventário dado.

**Exercício 7.** Crie uma função chamada `imprime_produto` que, dado um produto, imprime na tela seu código, nome, preço original e preço depois do desconto. Fique à vontade para criar a visualização que quiser, mas tente manter em uma só linha.

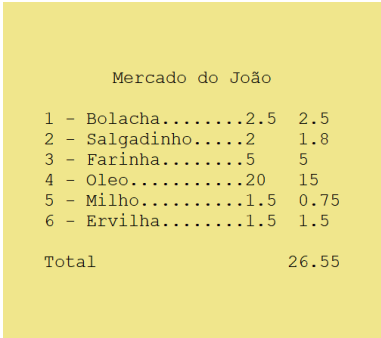
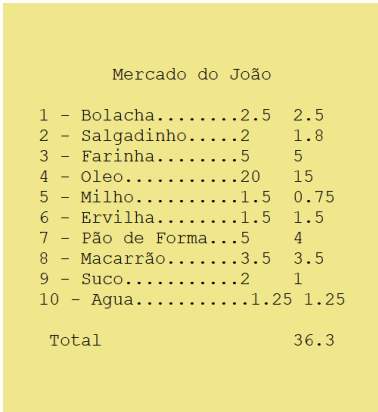
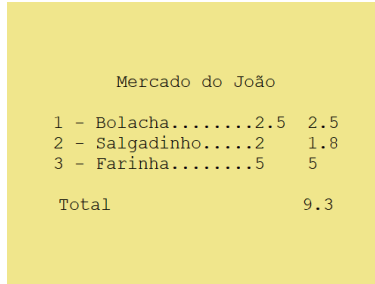
**Obs.:** A saída da função deve ser uma imagem com essas informações.

**Dica 1:** Se quiserem controlar melhor os tamanhos das strings, usem uma fonte em que todos os caracteres tem o mesmo tamanho, como a “courier”. Pra modificar a fonte utilizada, usem a função `text/font` do Racket.

**Exercício 8.** Crie uma função chamada `nota_fiscal` que, dado um carrinho de compras e um inventário, imprime a nota fiscal da compra. Pode assumir que todos os códigos contidos no carrinho constam no inventário dado. Você pode ser criativo para gerar a nota fiscal, mas ela deve obedecer às seguintes regras:

- conter todos os produtos do carrinho;
- cada produto deve ter as informações geradas pelo exercício anterior;
- conter o valor total da compra;
- a imagem deve ter um fundo retangular colorido;
- o tamanho da nota fiscal deve variar conforme a quantidade de produtos comprados.

Exemplo de notas fiscais geradas por compras de tamanhos diferentes:

 <pre> Mercado do João  1 - Bolacha.....2.5  2.5 2 - Salgadinho.....2  1.8 3 - Farinha.....5    5 4 - Oleo.....20     15 5 - Milho.....1.5    0.75 6 - Ervilha.....1.5   1.5  Total                26.55 </pre>	 <pre> Mercado do João  1 - Bolacha.....2.5  2.5 2 - Salgadinho.....2  1.8 3 - Farinha.....5    5 4 - Oleo.....20     15 5 - Milho.....1.5    0.75 6 - Ervilha.....1.5   1.5 7 - Pão de Forma...5    4 8 - Macarrão.....3.5   3.5 9 - Suco.....2       1 10 - Agua.....1.25   1.25  Total                36.3 </pre>	 <pre> Mercado do João  1 - Bolacha.....2.5  2.5 2 - Salgadinho.....2  1.8 3 - Farinha.....5    5  Total                9.3 </pre>
(a) NF com 6 produtos	(b) NF com 10 produtos	(c) NF com 3 produtos

**Exercício extra.** Refaça o exercício 6, porém agora sem assumir que todos os códigos constam no inventário. Se algum item não está no inventário, ele não pode ser comprado. Além do valor

total da compra, a função deve retornar um carrinho com os códigos dos itens que a pessoa conseguiu comprar. O nome da função deve ser `checkout_v2`

**Obs.:** Uma função em Racket só pode retornar um valor. Para retornar o carrinho e o número, construa uma estrutura chamada `compra` que contém um carrinho e um total (com esses nomes).