



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica 2023

Aula 01 – Introdução

Cuidados

- Os tipos, os nomes, e os parâmetros das funções **devem seguir o especificado** em cada exercício para fins de correção automática.
- A função `main` **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*. **Comente-a antes de submeter**.
- Use o código fornecido no e-Disciplinas.

Exercício 1

Implemente a função:

```
double calculaTaxaDeEntrega(int tipoDePagamento, bool temCupom, int distancia);
```

que calcula a taxa de entrega de um aplicativo de delivery. Essa função recebe três parâmetros: (i) `int tipoDePagamento`, que indica qual é a forma de pagamento que o cliente deseja utilizar; (ii) `bool temCupom`, uma variável booleana que se assumir o valor *true* indica que o usuário possui um cupom de frete grátis (*false* indica que não); e (iii) `int distancia`, que indica a quantidade de quilômetros percorridos pelo entregador.

Os tipos de pagamento disponíveis são os seguintes:

- 1 - Pix
- 2 - Ticket Alimentação
- 3 - Cartão de Crédito

O cupom de **frete grátis** só deve ser aplicado caso o método de pagamento seja **cartão de crédito (3)**. Usando Pix(1) ou Ticket (2) o cupom deve ser desconsiderado. Além disso, o cupom só é válido para **distâncias inferiores ou iguais a 5 quilômetros**.

- Caso o cliente não esteja elegível para o frete grátis, a taxa de entrega é calculada com R\$ 1,50 por quilômetro percorrido (por exemplo, 2 Km seria R\$ 3).
- Caso o método seja Pix (1) e a distância superior a 5 quilômetros há um desconto de 5% no valor da taxa de entrega.
- Caso a distância seja muito curta ($\text{distância} < 1$) o frete é grátis

Exemplo: `calculaTaxaDeEntrega(3, true, 4)` deve retornar 0
`calculaTaxaDeEntrega(3, false, 10)` deve retornar 15

O juiz eletrônico não verificará distâncias negativas nem outros tipos de pagamento.



Exercício 2

Implemente a função:

```
bool produtoEstaDisponivel(string produtos[], int quantidadeDisponivel[],  
    int tamanhoDaLista, string produtoBuscado, int quantidadeBuscada);
```

O vetor `produtos[]` contém a lista de todos os produtos disponíveis na loja virtual. Além disso, o vetor de `quantidadeDisponivel[]` indica o estoque de cada um dos produtos correspondentes armazenados em `produtos[]` (ou seja, a quantidade do produto na posição 0 de `produtos[]` está na posição 0 de `quantidadeDisponivel[]`; a quantidade do produto na posição 1 de `produtos[]` está na posição 1 de `quantidadeDisponivel[]` e assim por diante). O inteiro `tamanhoDaLista` indica o tamanho desses vetores. O string `produtoBuscado` indica o produto desejado pelo cliente e `quantidadeBuscada` indica a quantidade desejada deste produto. A função deve retornar `true` quando o produto for encontrado e a loja possui estoque suficiente. Caso contrário, ela deve retornar `false`.

Faça a comparação de strings usando “==”.

Por exemplo, no seguinte caso:

```
int tamanhoDaLista = 2  
string produtos[] = {"Arroz", "Feijao" }  
int quantidadeDisponivel[] = {10, 5}
```

A quantidade disponível de “Arroz” é 10 e de “Feijao” é 5. Então:

```
produtoEstaDisponivel(produtos, quantidadeDisponivel, 2, “Arroz”, 8)  
deve retornar true; e
```

```
    produtoEstaDisponivel(produtos, quantidadeDisponivel, 2, “Feijao”, 6)  
deve retornar false.
```

Exercício 3

Implemente a seguinte função:

```
double calcularPrecoTotal(string produtos[], double precos[],  
    int quantidadeDeProdutos, string produtosNaSacola[],  
    int quantidadeDeProdutosNaSacola);
```

O vetor `string produtos[]` e `double precos[]` possuem tamanho `int quantidadeDeProdutos`. O vetor de `precos[]` indica o preço de cada um dos produtos correspondentes armazenados em `produtos[]`. Dessa forma, eles possuem o mesmo tamanho. Por exemplo:

```
int quantidadeDeProdutos = 2  
string produtos[] = {"Arroz", "Feijao" }  
double precos[] = {20.0, 10.0}
```



Isso indica que o preço de “Arroz” é 20.0 e o de “Feijao” é 10.0.

De modo análogo, o vetor `produtosNaSacola[]` e o inteiro `quantidadeDeProdutosNaSacola` informam os produtos que o cliente deseja comprar. Por exemplo

```
quantidadeDeProdutosNaSacola = 3  
string produtosNaSacola[] = {“Arroz”, “Batata”, “Feijao” }
```

indica que o cliente solicitou “Arroz”, “Batata” e “Feijao”.

A função deve retornar a soma dos preços dos produtos na sacola. Caso o nome do produto esteja na sacola, mas não esteja no vetor `produtos`, ele deve ser desconsiderado. Com esses parâmetros a função deve retornar (em `double`) 30.0 - isso porque o “Arroz” custa 20.0, “Batata” não está em `produtos[]` e “Feijao” custa 10.0.

Testes do Judge

Exercício 1

- Cupom elegível
- Cupom não elegível
- Desconto por pix
- Sem cupom e distâncias curtas
- Sem cupom e distâncias longas
- Distância muito curta

Exercício 2

- Produto no início do vetor
- Produto no meio do vetor
- Produto no fim do vetor
- Produto indisponível em qualquer posição do vetor
- Produto disponível no fim do vetor

Exercício 3

- Catálogo e pedido simples
- Catálogo e pedido complexo
- Pedido parcialmente indisponível
- Pedido totalmente indisponível
- Pedido totalmente disponível