



Universidade Federal de Sergipe - UFS
Departamento de Sistemas de Informação - Itabaiana - DSI/Ita

Programação II - SINF0064

Exercício 05

Passagens de Parâmetros por Valor e por Referência

Prof. Dr. Alcides Xavier Benicasa

Exercício **AVALIATIVO**

Formato: *código-fonte*

Envio: encaminhar arquivos via **SIGAA**

Assunto: EX05 - Passagens de Parâmetros por Valor e por Referência

Passagem de Parâmetro por Valor

OBS.: não utilizar variável global.

1. Uma garagem de estacionamento cobra **R\$2,00** de **taxa mínima** para estacionar até três horas. A garagem cobra um adicional de **R\$0,50** por hora ou fração de **30 minutos** caso sejam excedidas as três horas. A taxa máxima para qualquer período determinado de 24 horas é **R\$10,00**. Admita que nenhum carro fique estacionado por mais de 24 horas. Escreva um programa que **calcule e imprima as taxas de estacionamento para três clientes** que estacionaram ontem seus carros nessa garagem. Você deve fornecer as horas que cada cliente ficou estacionado. Seu programa deve imprimir os resultados organizados conforme abaixo e deve calcular e imprimir o total recebido no dia de ontem. O programa deve usar a função **calculaTaxas** para determinar o valor a ser cobrado de cada cliente. A saída de seu programa deve ter o seguinte formato:

```
Forneça o tempo de permanência do primeiro cliente: 1.5
Forneça o tempo de permanência do segundo cliente: 4.5
Forneça o tempo de permanência do terceiro cliente: 24
Carro 1: 1.5hrs de permanência, Taxa de R$ 2
Carro 2: 4hrs de permanência, Taxa de R$ 2.75
Carro 3: 24hrs de permanência, Taxa de R$ 10
TOTAL: Horas[30] / Taxas[R$ 14.75]
Outra vez? (s/n):
```

Inclua um *loop* que permita ao usuário repetir esse cálculo para novos dados de entrada até o usuário dizer que deseja encerrar o programa.

2. Implemente as seguintes funções:
 - a) A função **celsius** retorna a temperatura em **Celsius** equivalente a uma temperatura em Fahrenheit.
 - b) A função **fahrenheit** retorna a temperatura em **Fahrenheit** equivalente a uma temperatura em Celsius.
 - c) Use essas funções para escrever um programa que mostre as temperaturas Fahrenheit equivalentes às temperaturas Celsius de 0 à 100 graus e as temperaturas Celsius equivalentes às temperaturas Fahrenheit de 32 à 212 graus. Apresenta a saída de forma legível.

3. Escreva uma **função** que receba a hora como três argumentos inteiros (para **horas**, **minutos** e **segundos**) e retorne o número total de segundos. Use essa função para calcular o **intervalo de tempo em segundos entre duas horas** e também o **intervalo de tempo em horas entre essas duas horas**, ambas dentro de um ciclo de doze horas do relógio.

Passagem de Parâmetro por Referência

OBS.: não utilizar variável global e/ou função que retorne valor.

1. A área de um triângulo arbitrário pode ser calculada por meio da fórmula:

$$area = \sqrt{s(s-a)(s-b)(s-c)} \quad (1)$$

onde a , b , e c são as medidas dos lados e s é o semiperímetro, definido a seguir:

$$s = (a + b + c)/2 \quad (2)$$

Escreva uma função *void* que utilize cinco parâmetros: três parâmetros chamados por valor que forneçam as medidas dos lados e dois parâmetros chamados por referência que calculem a área e o perímetro (**não o semiperímetro**).

2. Escreva um programa que diga quantas moedas retornar para qualquer quantia de 1 a 99 reais. Por exemplo, se a quantia é 86 centavos, a saída deve ser algo parecido com:

86 reais podem ser fornecidos como

3 de 25 centavo(s), 1 de 10 centavo(s) e 1 de 1 centavo(s)

Utilize denominações para moedas de 25 centavos, 10 centavos e 1 centavo. Não utilize moedas de 50 centavos nem de 5 centavos.

Seu programa utilizará a seguinte função (entre outras):

```
void calculaMoedas(int valorDaMoeda, int& numero, int& quantiaRestante);
```

```
// Pré-condição: 0 < valorDaMoeda < 100; 0 <= quantiaRestante < 100.
```

```
// Pós-condição: número fixado como igual ao número máximo
```

```
// de moedas de denominação valorDaMoeda centavos que possa ser obtido
```

```
// a partir de quantiaRestante centavos. quantiaRestante diminui conforme
```

```
// o valor das moedas, ou seja, diminui o número*valorDaMoeda.
```

Por exemplo, suponha que o valor da variável *quantiaRestante* seja 86. Então, depois da seguinte chamada, o valor de *numero* será 3 e o valor da *quantiaRestante* será 11 (porque se você retira 75 de 86, restam 11):

```
calculaMoedas(25, numero, quantiaRestante);
```

Inclua um loop que permita ao usuário repetir esse cálculo para novos dados de entrada até o usuário dizer que deseja encerrar o programa. (*Dica: utilize divisão de inteiros e o operador % para implementar essa função*).

3. Escreva um programa que converta da notação de 24 horas para a notação de 12 horas. Por exemplo, o programa deve converter 14:25 em 2:25 P.M. A entrada é dada em dois inteiros. Deve haver pelo menos três funções: uma para a entrada, uma para fazer a conversão e uma para a saída. Registre a informação A.M./P.M. como um valor do tipo **char**, 'A' para A.M. e 'P' para P.M. Assim, a função para efetuar as conversões terá um parâmetro formal chamado por referência de tipo **char** para registrar se é A.M. ou P.M. (a função terá outros parâmetros também). Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar, até o usuário dizer que deseja encerrar o programa.