

Linguagem de Programação



04: ESTRUTURAS CONDICIONAIS (continuação)

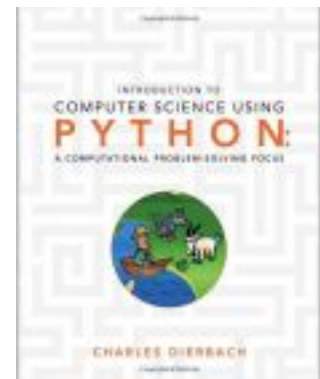
Nossos objetivos nesta aula são:

- Saber o que é uma estrutura de controle de fluxo.
- Conhecer os operadores relacionais e os operadores lógicos.
- Desenvolver e avaliar o resultado de expressões lógicas envolvendo operadores relacionais e lógicos.
- Construir algoritmos com desvio de fluxo.
- Utilizar os principais comandos condicionais: simples, composto, aninhado e encadeado.
- Ser capaz de ler, escrever e implementar programas que utilizem estruturas condicionais.



A referência para esta aula são as seções **3.1 (O que é uma estrutura de controle)**, **3.2 (Expressões lógicas)** e **3.3 (Controle de Seleção)** do **Capítulo 3 (Control Structures)** do livro:

DIERBACH, C. *Introduction to Computer Science Using Python: A Computational Problem Solving Focus*. 1st Edition, New York: Wiley, 2012.



MOTIVAÇÃO

Os primeiros computadores eletrônicos foram referidos como "cérebros eletrônicos". Deu-se a impressão enganosa de que os computadores poderiam "pensar". Embora possam ser complexos em seu *design*, computadores são máquinas que simplesmente fazem, passo a passo (instrução por instrução), o que lhes é informado.

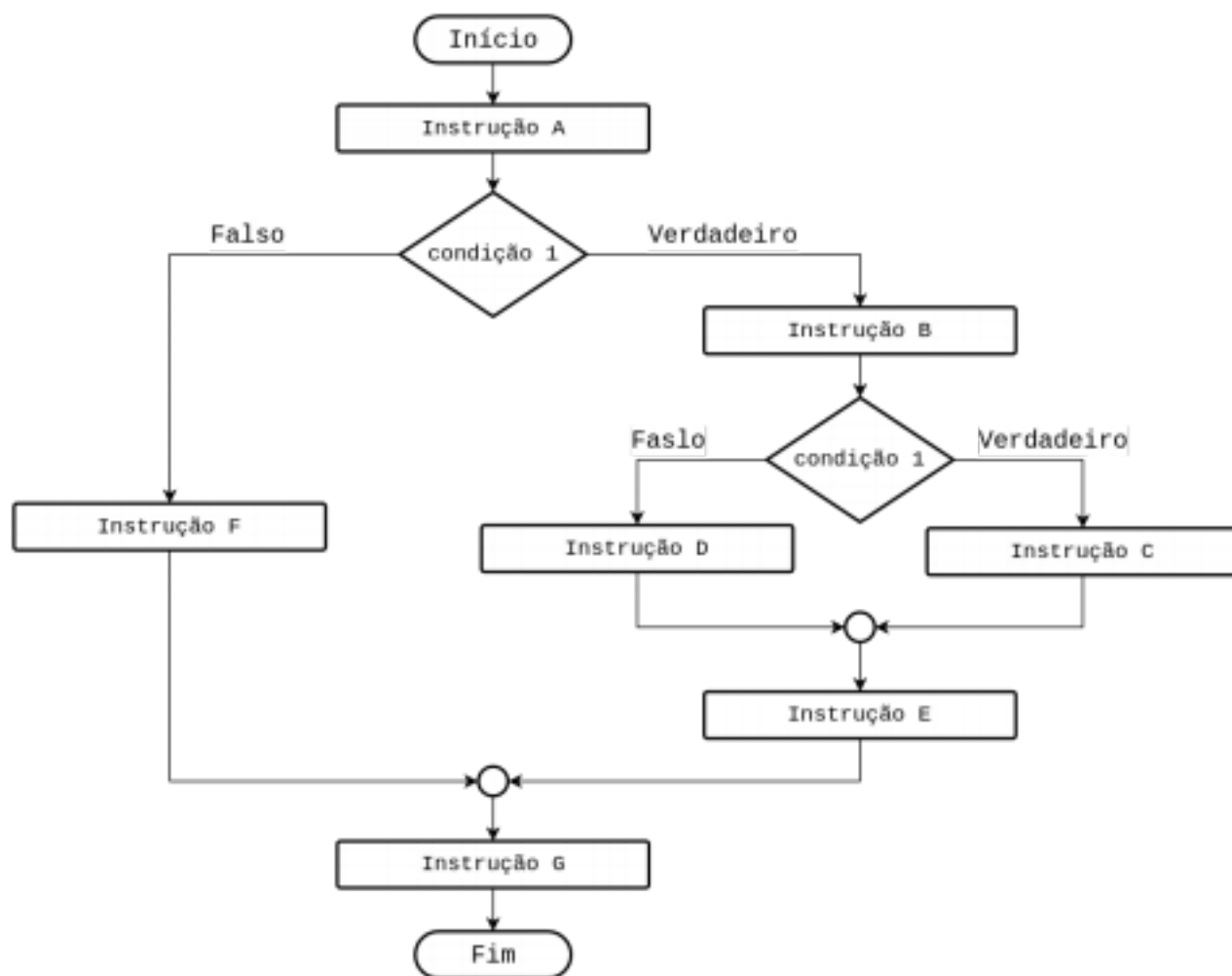


Assim, não há mais inteligência em um computador do que aquilo que ele é instruído a fazer. O que os computadores podem fazer, no entanto, é executar uma série de instruções muito rapidamente e de forma confiável.

É a velocidade em que as instruções podem ser executadas que dá aos computadores o seu poder, uma vez que a execução de muitas instruções simples pode resultar em comportamentos complexos. Vamos a partir de agora controlar não só a ordem em que as instruções são executadas, mas também se de fato serão executadas.

ESTRUTURA DE SELEÇÃO ANINHADA OU ENCADEADA

Tanto na estrutura de seleção simples quanto na seleção composta há apenas uma condição para ser avaliada, e que condicionará o fluxo de execução do programa. Portanto, há no máximo uma bifurcação para um caminho, entre dois possíveis. E quando é necessário resolver um problema que possui mais de dois caminhos possíveis? Para lidar com situações como essa, podemos usar estruturas de seleção aninhadas ou encadeadas, como no exemplo a seguir:



Dizemos que a estrutura de seleção da condição 2 está **aninhada** na condição 1, neste caso dentro do bloco verdadeiro, mas poderíamos ter também uma outra estrutura de seleção aninhada no bloco falso da condição 1. E podemos continuar aninhando estruturas de seleção quantas vezes for necessário para resolver o nosso problema. A seleção aninhada é portanto o agrupamento de uma ou mais seleções internas à uma seleção externa.

Existe ainda um caso específico de aninhamento que veremos mais adiante neste capítulo, quando a seleção interna está contida no bloco falso da seleção externa, e compõe todo o bloco falso, isto é, o único código dentro do bloco falso é o código da seleção interna. Quando isto acontece, dizemos que as estruturas de seleção estão **encadeadas**, e o Python possui um comando especial para esta situação que podemos chamar de “confeito sintático”, usado para simplificar a sintaxe.

Veja no trecho abaixo a tradução para Python do fluxograma dado no exemplo anterior:

<Instrução A - antes da seleção>

```
if <condição 1>:
    <Instrução B - início do bloco True da condição 1>
    if <condição 2>:
        <Instrução C - bloco True da condição 2>
    else:
        <Instrução D - bloco False da condição 2>
    <Instrução E - aqui ainda é o bloco True da condição 1>
else:
    <Instrução F - bloco False da condição 1>

<Instrução G - depois da seleção>
```

EXEMPLO DE APLICAÇÃO

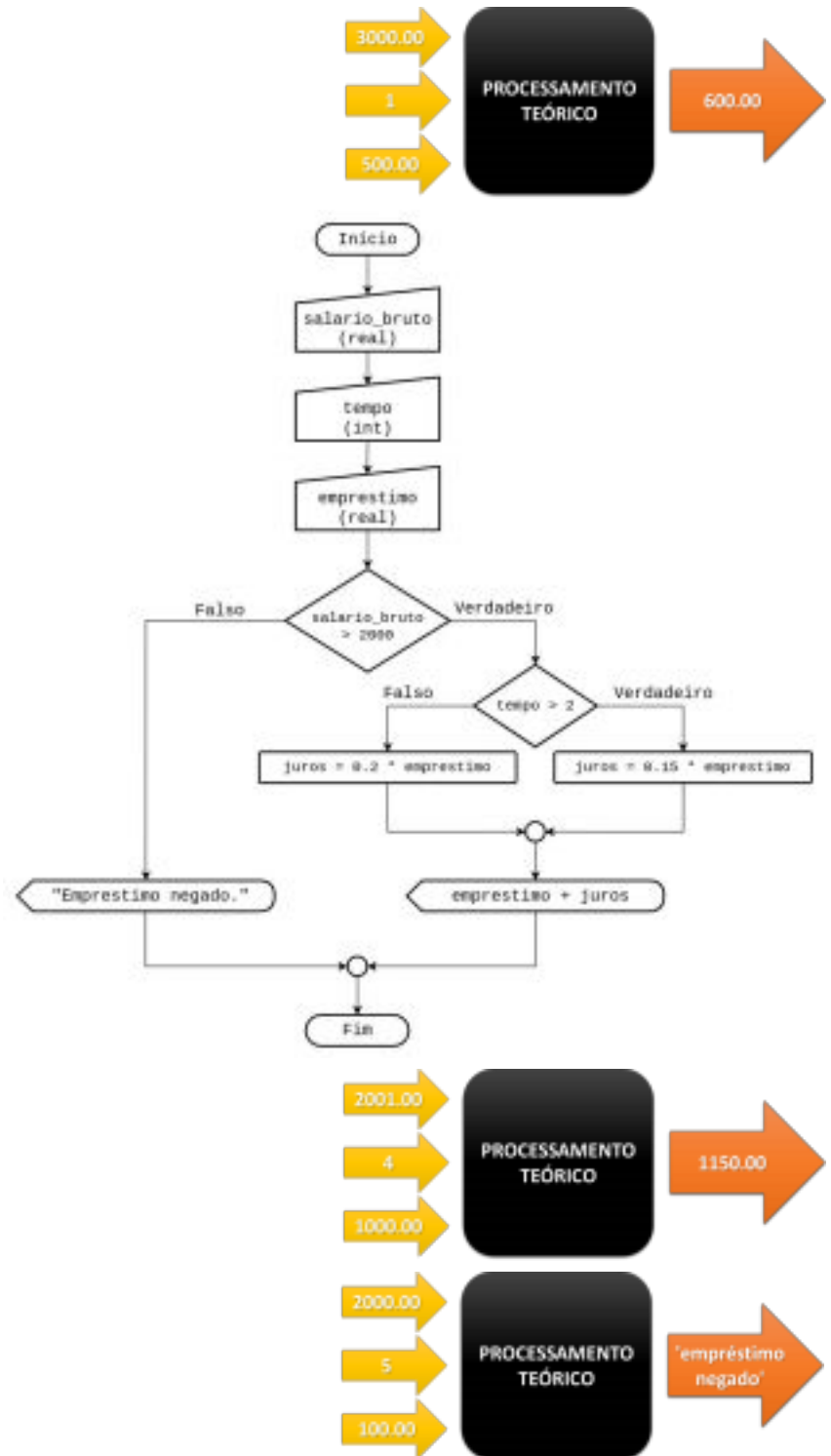
Considere a seguinte situação: um banco oferece uma modalidade de empréstimo de parcela única para qualquer cliente cujo salário bruto ultrapasse R\$ 2000,00. Essa modalidade também considera que caso o cliente tenha mais de dois anos de contrato, serão cobrados juros de 15% sobre o valor emprestado, caso contrário os juros serão de 20%. Se o salário bruto não atingir o mínimo estipulado, o empréstimo será negado. O gerente do banco quer um programa que receba como entrada o salário bruto (em reais e centavos), o tempo de contrato do cliente com o banco (em anos) e o valor solicitado de empréstimo (em reais e centavos), e exibe o valor da dívida do cliente, isto é, o valor emprestado acrescido dos juros.

Note que com apenas os recursos das estruturas de seleção simples e composta não seria simples modelar esta situação, pois o enunciado evidencia que existem mais de dois possíveis fluxos de execução no algoritmo, que dependem dos dados passados como entrada. Um exemplo de código que resolve esse problema é:

```
salario_bruto = float(input('Digite o salário: '))
tempo = int(input('Há quantos anos é cliente? '))
emprestimo = float(input('Digite o valor do empréstimo: '))

if salario_bruto > 2000:
    if tempo > 2:
        juros = 0.15 * emprestimo
    else:
        juros = 0.2 * emprestimo
    print(emprestimo + juros)
else:
    print('Empréstimo negado.')
```

Um fluxograma adequado para esse problema é ilustrado a seguir:



1° teste de mesa

salário

brutotempo empréstimo juros
salário bruto

> 2000tempo > 2 Tela

2000.0 5 100.0 -2000.0 > 2000

(False)-'Empréstimo
negado.'

2° teste de mesa

salário

brutotempo empréstimo juros
salário bruto

> 2000tempo > 2 Tela

3000.0 5 500.0 -3000.0 > 2000 (True)

brutotempo empréstimo juros
salário bruto
1 > 2

3° teste de mesa

(False)600.0

salário

> 2000tempo > 2 Tela

2001.0 5 1000.0 -2001.0 > 2000 (True)

4 > 2

(True)^{1150.0}

4

SELEÇÃO ENCADEADA - LEGIBILIDADE DO CÓDIGO

Note que o uso de seleção encadeada é útil e necessário em diversas situações, essencialmente quando os problemas que devem ser solucionados podem conter diversos valores diferentes, exigindo uma tomada de decisão para verificar qual a sequência de instruções que será feita de acordo com o valor dado.

Porém, quando há muitas seleções encadeadas, uma consequência da indentação necessária para o correto funcionamento do código é a perda de legibilidade, pois o código acaba ficando excessivamente deslocado para a direita.

Veja um exemplo do excesso de deslocamento para a direita em um programa com diversas seleções encadeadas, que recebe como entrada um número natural e exibe qual o respectivo dia da semana, considerando domingo como dia 1 e sábado como dia 7:

```
dia = int(input())

if dia == 1:
    print('domingo')
else:
    if dia == 2:
        print('segunda')
    else:
        if dia == 3:
            print('terça')
        else:
            if dia == 4:
                print('quarta')
            else:
                if dia == 5:
                    print('quinta')
                else:
                    if dia == 6:
                        print('sexta')
                    else:
                        print('sábado')
```

Como dito no começo do capítulo, cada bloco `else` do exemplo contém apenas uma estrutura de seleção composta, sem nenhum código que não faça parte dessa estrutura de seleção interna. Aqui não importa o que compõe o código dos blocos da seleção interna, que pode conter outras seleções aninhadas. Podemos portanto usar a estrutura do Python que combina o **else** com o **if** que vem logo em seguida, formando o comando **elif**.

```
dia = int(input())

if dia == 1:
    print('domingo')
elif dia == 2:
    print('segunda')
elif dia == 3:
    print('terça')
elif dia == 4:
    print('quarta')
elif dia == 5:
    print('quinta')
elif dia == 6:
    print('sexta')
else:
    print('sábado')
```

Note que não há diferença no encadeamento das condições, o fluxo executado é exatamente o mesmo em ambos os códigos, podendo inclusive ser representado pelo mesmo fluxograma.

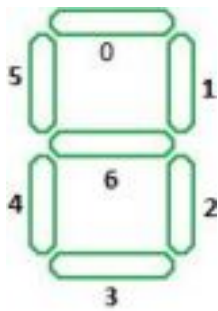
Vale ressaltar aqui que o problema usado de exemplo seria mais facilmente tratado em Python usando uma estrutura de dados diferente, como um dicionário, por exemplo, evitando assim a necessidade de uma estrutura de seleção para fazer essa associação entre números e dias da semana, mas tal estrutura não será vista neste momento do curso.

5

Para praticar - display de 7 segmentos

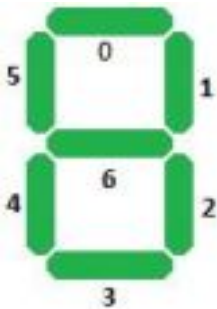
Vamos criar um programa que exiba um display de 7 segmentos com um dígito, usando *ASCII art*. Para isso escreva um programa que receba um número inteiro no intervalo [0, 9], e exiba na tela o número formado pelo display de 7 segmentos.

Observe na figura abaixo que cada segmento tem um número associado a

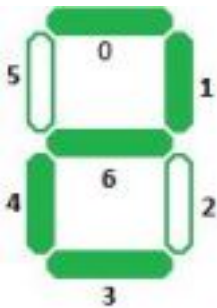


ele:

Display com todos os segmentos apagados



Display com todos os segmentos acesos



Display com os segmentos 0,1,3,4 e 6 acesos formando o número 2

EXEMPLO DE EXECUÇÃO DO CÓDIGO

Se quiser, após concluir o exercício como mostrado no exemplo a seguir, tente fazer outros padrões para formar os números, pesquise por “ASCII art” se precisar de uma inspiração.

```
>>> Digite um número entre 0 e 9: 5
```

—

|_
_|

6

EXERCÍCIOS PROPOSTOS (ESTRUTURA DE SELEÇÃO ANINHADA E ENCADEADA)

1) Considere o problema de conversão de temperatura com três possibilidades:

- i) Celsius para Fahrenheit (usuário deve digitar ' F ');
- ii) Fahrenheit para Celsius (usuário deve digitar ' C '); e
- iii) O usuário digitar um valor inválido (qualquer coisa diferente de ' C ' ou ' F '). Faça um programa que mostra a temperatura convertida ou uma mensagem de opção inválida.

2) Um posto de combustível vende três tipos de combustível: álcool, diesel e gasolina. O preço de cada litro de combustível é apresentado na tabela abaixo. Faça um programa que leia a quantidade de litros, um caractere que representa o tipo de combustível comprado (' a ' , ' d ' ou ' g ') e exibe o valor em reais a ser pago.

Combustível Preço por Litro (R\$)

a (álcool) 3,8997
d (diesel) 3,6543
g (gasolina) 4,4009

3) Um banco concederá um crédito especial aos seus clientes de acordo com o saldo médio no último ano. Crie um programa que recebe o saldo médio de um cliente, calcula e mostra o valor do crédito, de acordo com a tabela a seguir.

Saldo Médio Percentual do saldo médio

Acima de R\$ 4.000,00 30%
De R\$ 3.000,01 a R\$ 4.000,00 25%
De R\$ 2.000,01 a R\$ 3.000,00 20%
Até R\$ 2.000,00 10%

4) Escreva um programa que tenha como entrada o tipo do voo (' N ' para noturno e ' D ' para diurno) e a quantidade de pessoas; calcule e mostre a tarifa e o total a pagar de acordo com as condições abaixo:

Tipo de Voo Número de pessoas Tarifa por pessoa

Diurno 50 ou menos R\$ 200,00
mais que 50 R\$ 120,00

Noturno⁵⁰ ou menos R\$ 100,00

mais que 50 R\$ 80,00

7

5) Faça um programa que exhibe o valor de $f(x)$ de acordo com a entrada x .

$$f(x) = \begin{cases} 1 & \text{se } x \leq 1 \\ 2 & \text{se } 1 < x \leq 2 \\ x^2 & \text{se } 2 < x \leq 3 \\ x^3 & \text{se } x \geq 3 \end{cases}$$

6) Uma das métricas que um endocrinologista pode usar para averiguar a saúde de seus pacientes é o Índice de Massa Corpórea (IMC). Sabe-se que o IMC é calculado através da seguinte fórmula: $\text{IMC} = \text{peso}/\text{altura}^2$, em que o peso é dado em quilos e altura é dada em metros. Escreva um programa que solicita o peso e a altura de um usuário e apresenta o IMC da pessoa e sua faixa de risco, com base na seguinte tabela:

IMC Faixa de risco

IMC < 20 Abaixo do peso

20 <= IMC <= 25 Normal

25 < IMC <= 30 Excesso de peso

30 < IMC <= 35 Obesidade

IMC > 35 Obesidade mórbida

7) Escreva um programa que recebe dois números e mostra o resultado das operações listadas a seguir de acordo com a escolha do usuário:

Opção Operação

1 Média entre os números digitados

2 Diferença do maior pelo menor

3 Produto entre os números digitados

4 Divisão do primeiro pelo segundo

Obs.: Se a opção digitada for inválida, mostrar uma mensagem de erro e encerrar a execução. Lembre-se de que, na opção 4, o segundo número deve ser diferente de zero.

8) Um hospital precisa de um programa para calcular e imprimir os gastos de um paciente. A tabela de preços do hospital é a seguinte:

- Quartos:
 - Particular: R\$ 360,00
 - Semi-particular: R\$ 210,00
 - Coletivo: R\$ 185,00
- Internet: R\$ 3,00
- TV a cabo: R\$ 4,00

Escreva um programa que leia o número de dias gastos no hospital, o tipo de quarto, se usou ou não o WIFI (Sim, Não), e se usou ou não a TV a cabo (Sim, Não).

8

Então emita um relatório, como o do exemplo a seguir:

```
Hospital Comunitário
Dias no hospital: 5
Tipo de quarto...: Particular
Diárias.....: R$ 1800.00
Internet.....: R$ 3.00
TV a cabo.....: R$ 4.00
Total.....: R$ 1807.00
```

DICA: Para formatar um número com um tamanho fixo, use a seguinte estrutura em

Python: `'R$ {:X.Y}'.format(valor)`

Onde X é o número total de dígitos, incluindo o ponto, e Y é o número de casas decimais que se deseja exibir. Para gerar o texto do exemplo, podemos fazer:

```
>>> 'R$ {:7.2}'.format(3)
'R$ 3.00'
```

9) Faça um programa que receba como entrada o código da região de localização do cliente, o nome do cliente, o número de peças vendidas e o nome do vendedor e exiba o valor do frete, a comissão do vendedor e o lucro obtido com a venda. Sabendo-se que:

- O valor do frete depende da quantidade transportada e da região;
- A comissão do vendedor é de 6,5 % do valor total da venda;
- Lucro: Valor total da venda - custo total - comissão do vendedor;
- Custo por peça: R\$ 7,00;
- Custo total: custo por peça × número de peças vendidas;
- Valor total da venda: custo total acrescido em 50%;
- Valor do frete por região:

Região Valor do frete

<u>Código</u>	<u>Nome</u>	<u>Até 1.000 peças (R\$ /</u>	<u>Acima de 1.000 peças (% do custo</u>
		<u>peça)</u>	<u>total)</u>
1	Sul	1,00 10%	2 Norte 1,10 8%
3	Leste	1,15 7%	4 Oeste 1,20 11%
5	Noroeste	1,25 15%	6 Sudeste 1,30 12%
7	Centro-Oeste	1,40 18%	
8	Nordeste	1,35 15%	

9

10) Faça um programa que calcula o valor total a ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

Códigos Condições de pagamento

- 1 À vista em dinheiro ou cheque, recebe 10% de desconto
- 2 À vista no cartão de crédito, recebe 5% de desconto
- 3 Em 2 vezes, preço normal de etiqueta sem juros
- 4 Em 3 vezes, preço normal de etiqueta mais juros de 10%

TAREFA MÍNIMA (ESTRUTURAS DE SELEÇÃO ANINHADA E ENCADEADA)

Não se esqueça, conseguir resolver esses exercícios sem muitas dificuldades é um importante parâmetro para saber como estão seus conhecimentos sobre o assunto do capítulo.

Exercício 1 - Resolvido

Refaça o programa de nosso primeiro exemplo, transformando-o em uma função. Não se esqueça de fazer os testes de caixa-preta e de caixa-branca (usando teste de mesa) para verificá-lo. Por último, faça outra versão do programa, desta vez com o salário bruto, tempo de contrato e valor do empréstimo passados como parâmetros.

RESOLUÇÃO ✓



sb	tempo	emp	juros	sb > 2000.00	tempo > 2	Tela
1000.00	7	100.00	-	1000.00 > 2000.00 (False)	-	empréstimo negado'



11



Exercício 2

Crie um programa que leia três números inteiros e exiba apenas o maior deles. Em seguida faça outro programa do mesmo problema, porém em versão função sem parâmetros. Por último, crie uma terceira versão que use uma função que receba os três números como parâmetros. Não esqueça de fazer testes de caixa-preta e de caixa-branca (usando teste de mesa).

Exercício 3 - Com dicas

Faça um programa que leia três números inteiros e distintos e exiba-os em ordem crescente. Não esqueça de fazer testes de caixa-preta e de caixa-branca (usando teste de mesa).

12

DICAS

Comece comparando se o primeiro número é o maior de todos, caso seja exiba-o e verifique qual é o segundo maior, consequentemente o terceiro será descoberto também. Caso o primeiro número não seja o maior de todos, verifique se o segundo é o maior de todos e siga a mesma estratégia anterior para descobrir quais são o segundo e o terceiro. Por fim, caso o segundo também não seja o maior de todos, com certeza o maior é o terceiro, então repita a mesma estratégia anterior para descobrir quais são o segundo e o terceiro.

Exercício 4

Faça um programa que lê três valores reais (**a**, **b** e **c**), representando os coeficientes de uma equação do 2º grau, e exibe o valor de Δ e os valores das raízes reais, caso existam. Considere:

- se **a** for zero, exiba a mensagem 'não é equação de 2º grau' e encerre;
- se Δ for negativo, exiba a mensagem 'não existem raízes reais' e encerre;
- Se Δ for zero ou positivo, exiba a(s) raiz(es) e encerre.

Em seguida faça outro programa do mesmo problema, porém em versão função. Não esqueça de fazer testes de caixa-preta e de caixa-branca (usando teste de mesa). Por fim, construa uma terceira versão, em que a função recebe os valores de **a**, **b** e **c** como parâmetros.

Exercício 5 - Com dicas

Faça um programa que faz a leitura de três valores reais (**a**, **b** e **c**), representando o comprimento de três retas, e exibe uma mensagem indicando se é possível ligar as pontas das retas formando um triângulo (pesquise qual a *condição de existência* de um triângulo). Caso seja possível formá-lo, indique qual o seu tipo (equilátero, isósceles ou escaleno). Faça o teste de caixa-preta e caixa-branca usando testes de mesa. Ao final, faça mais duas versões do programa, uma que usa uma função sem parâmetros e outra em que os valores **a**, **b** e **c** são passados como parâmetros.

DICAS

Triângulo é uma forma geométrica composta de três lados, onde cada lado é menor que a soma dos outros dois lados. Perceba que isto é uma regra (uma condição) e deverá ser considerada, isto é, só é possível formar um triângulo quando temos:

$$(a < b+c) \text{ E } (b < a+c) \text{ E } (c < a+b)$$

Uma vez confirmado que os valores informados para os três lados formam um triângulo, faça as comparações necessárias para estabelecer qual é o tipo do triângulo formado: ■ **Equilátero**: todos os lados são iguais;

■ **Isósceles**: somente dois lados iguais;

■ **Escaleno**: todos os lados são diferentes.