

# Aprenda e Melhore a Sua Lógica de Programação com Java



## Lista de Exercícios

JAN/2023

João Ricardo Côre Dutra



## Sumário

Sobre o curso .....	2
Fundamentação teórica .....	3
Instrução escrever .....	3
Instrução ler .....	3
Variáveis .....	4
Operadores de atribuição .....	4
Operadores aritméticos .....	6
Operadores relacionais .....	7
Operadores lógicos .....	8
Estrutura condicional - IF .....	8
Estrutura condicional ou de seleção - SWITCH CASE .....	10
Estrutura de repetição - WHILE .....	11
Estrutura de repetição - DO WHILE .....	11
Estrutura de repetição - FOR .....	12
Função .....	12
Vetores .....	13
Exercícios de estrutura sequencial .....	15
Exercícios de estrutura de decisão .....	17
Exercícios de estrutura de repetição .....	23
Exercício com arquivos .....	33
Exercícios com strings .....	33
Exercícios com listas .....	37
Exercícios com funções .....	45
Exercícios com classes .....	48



## Sobre o curso

A habilidade de resolver problemas através de algoritmos é fundamental para um programador no seu dia-a-dia e deve anteceder ao conhecimento de frameworks modernos como spring, quarkus, react.js, angular, django, flask e muitos outros. Você já parou para pensar que um atleta de alto desempenho pratica o esporte que compete até à exaustão para se tornar realmente bom? Assim como um pianista de sucesso treina tocar piano massivamente desde muito cedo para se tornar um artista de renome? O mesmo deve ocorrer com um programador que deve praticar lógica de programação até se tornar realmente bom em programar com qualquer linguagem e/ou framework.

Estou muito animado por apresentar o curso "Algoritmos Básicos Com JAVA: Aprenda através da Resolução Massiva de Exercícios". Este curso tem como objetivo aprimorar sua habilidade em construir algoritmos através da resolução de mais de 100 exercícios cedidos pelo Professor Marco André Mendes e disponibilizados no site Python Brasil (<https://wiki.python.org.br/ListaDeExercicios>). Aproveito a oportunidade para agradecer ao Professor Marco André Mendes pelo apoio e pela disponibilização deste valioso material.

O curso será ministrado por mim (João Ricardo Côre Dutra) onde o repositório oficial do Github encontra-se no link <https://github.com/jrdutra/algoritmos>. Caso queira a resolução em vídeo destes exercícios, este curso está disponível para aquisição a um preço simbólico no endereço <https://udemy.com.br>.

Bons estudos!



## Fundamentação teórica

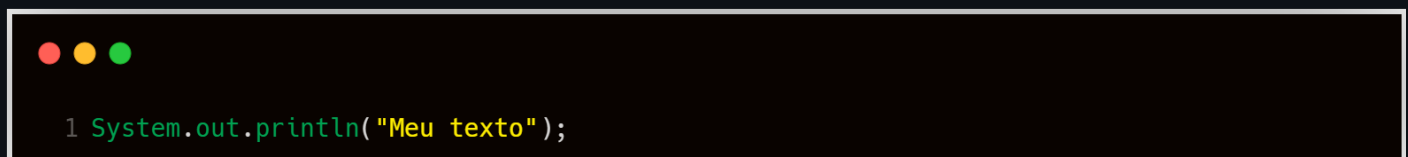
Nesta primeira etapa do curso, você aprenderá as habilidades básicas e conceitos fundamentais para começar a programar. Você descobrirá como pensar como um programador, como formular soluções para problemas e como traduzi-los em código. Aprenderá sobre estruturas de controle de fluxo, variáveis, funções e outros conceitos importantes da programação. Além disso, você terá a oportunidade de praticar suas habilidades através de exercícios práticos e desafios. Então, se prepare para uma jornada divertida e desafiadora na lógica de programação!

### Instrução escrever

A instrução `System.out.println('Meu texto')` é uma instrução de escrita na linguagem Java. Ela é usada para exibir uma mensagem (neste caso, "Meu texto") na saída padrão (geralmente o console).

Em outras palavras, a instrução `System.out.println('Meu texto')` exibe a mensagem "Meu texto" na tela para o usuário ver. É uma forma simples e direta de exibir informações na saída padrão durante a execução de um programa Java.

Exemplo de uso:



### Instrução ler

O Scanner é uma classe do Java que permite ler dados de fontes como o teclado, arquivos, entre outros. Ele é usado para ler informações digitadas pelo usuário do teclado. Para ler informações do teclado, é necessário criar um objeto Scanner e associá-lo ao objeto `System.in`, que representa a entrada padrão do sistema (teclado). Em seguida, usamos os métodos do Scanner para ler os dados, como `nextInt()`, `nextLine()` ou `nextDouble()`.



Exemplo de uso:

```
1 Scanner ler = new Scanner(System.in);  
2 int a = ler.nextInt();
```

Neste exemplo, criamos um objeto Scanner chamado sc e associamos a System.in. Em seguida, usamos o método nextInt() para ler um inteiro digitado pelo usuário.

## Variáveis

Variáveis são nomes atribuídos a espaços de memória que armazenam valores. Em programação, variáveis são usadas para armazenar dados temporariamente durante a execução de um programa. Cada variável tem um tipo, como inteiro, ponto flutuante, string, etc., que determina o tipo de dado que pode ser armazenado nela. As variáveis podem ser lidas, modificadas e usadas em operações matemáticas e lógicas.

Exemplo de uso:

```
1 int numeroA = 1;  
2 int numeroB = 3;  
3 float numeroDecimal = 1.0;  
4 char catactere = 'a';  
5 boolean existeAlgo = true;  
6 String meuTexto = "Olá mundo!";
```

## Operadores de atribuição

Os operadores de atribuição da linguagem Java permitem que você atribua valores a variáveis. Eles são uma parte importante da programação em Java e são usados com frequência em muitos códigos. Aqui estão os operadores de atribuição mais comuns em Java:



1. Atribuição simples (=): Este operador é usado para atribuir um valor a uma variável. Por exemplo, `int x = 10;` atribui o valor 10 à variável x.
2. Adição e atribuição (+=): Este operador adiciona um valor a uma variável existente e atribui o resultado de volta à variável. Por exemplo, `x += 5;` adiciona 5 ao valor existente de x e atribui o resultado a x.
3. Subtração e atribuição (-=): Este operador subtrai um valor de uma variável existente e atribui o resultado de volta à variável. Por exemplo, `x -= 5;` subtrai 5 do valor existente de x e atribui o resultado a x.
4. Multiplicação e atribuição (\*=): Este operador multiplica um valor por uma variável existente e atribui o resultado de volta à variável. Por exemplo, `x *= 5;` multiplica x por 5 e atribui o resultado a x.
5. Divisão e atribuição (/=): Este operador divide um valor por uma variável existente e atribui o resultado de volta à variável. Por exemplo, `x /= 5;` divide x por 5 e atribui o resultado a x.
6. Resto e atribuição (%=): Este operador calcula o resto de uma divisão de um valor por uma variável existente e atribui o resultado de volta à variável. Por exemplo, `x %= 5;` calcula o resto da divisão de x por 5 e atribui o resultado a x.

Podemos perceber que os operadores de atribuição em Java são uma ferramenta poderosa para manipular e atribuir valores a variáveis de maneira eficiente. Eles são amplamente utilizados na programação em Java e são uma habilidade importante para qualquer programador iniciante aprender.

Exemplo de uso:

```
1 int numeroA = 1;
2 int numeroB += 2;
3 numeroB += numeroA;
4 int numeroC = numeroB;
```



## Operadores aritméticos

Os operadores aritméticos são utilizados para realizar operações matemáticas em um programa. Os operadores aritméticos da linguagem Java são:

1. Adição (+): Adiciona dois valores e retorna o resultado
2. Subtração (-): Subtrai o segundo valor do primeiro e retorna o resultado
3. Multiplicação (\*): Multiplica dois valores e retorna o resultado
4. Divisão (/): Divide o primeiro valor pelo segundo e retorna o resultado
5. Módulo (%): Retorna o resto da divisão inteira do primeiro valor pelo segundo

Exemplo de uso:

```
1 int a = 10;  
2 int b = 5;  
3 int c = a + b;  
4 int d = a - b;  
5 int e = a * b;  
6 int f = a / b;  
7 int g = a % b;
```



## Operadores relacionais

Os operadores relacionais em Java são utilizados para comparação de valores e retornam um valor booleano (verdadeiro ou falso). Eles são:

1. Igual a (==): verifica se dois valores são iguais.
2. Diferente de (!=): verifica se dois valores são diferentes.
3. Maior que (>): verifica se o valor da esquerda é maior do que o valor da direita.
4. Menor que (<): verifica se o valor da esquerda é menor do que o valor da direita.
5. Maior ou igual a (>=): verifica se o valor da esquerda é maior ou igual ao valor da direita.
6. Menor ou igual a (<=): verifica se o valor da esquerda é menor ou igual ao valor da direita.

Exemplo de uso:

```
1 idade > 18
```





## Operadores lógicos

Os operadores lógicos em Java são usados para combinar expressões booleanas e retornam um valor booleano. Aqui estão os principais operadores lógicos em Java:

1. Operador E lógico (&&): retorna verdadeiro se ambas as expressões booleanas forem verdadeiras.
2. Operador OU lógico (||): retorna verdadeiro se pelo menos uma das expressões booleanas for verdadeira.
3. Operador NÃO lógico (!): inverte o valor booleano de uma expressão.
4. Operador condicional (? :) : permite testar uma expressão condicional e retornar um valor específico de acordo com o resultado.

Esses operadores são utilizados em estruturas de controle de fluxo como if-else, while, for, etc. para tomar decisões baseadas em expressões booleanas. Eles são fundamentais na programação orientada a objetos e na construção de programas complexos.

Exemplo de uso:

```
1 idade > 18 && idade < 60
```

## Estrutura condicional - IF

Estruturas condicionais são blocos de código que permitem executar diferentes ações de acordo com uma determinada condição. Em outras palavras, a execução do código dentro da estrutura condicional depende se a condição especificada é verdadeira ou falsa. Em muitas linguagens de programação, a estrutura condicional mais comum é o "if-else". O "if" especifica a condição e o "else" especifica o que acontece se a condição não for satisfeita.



Exemplo de uso:

```
1      int a = 1;
2      int b = 2;
3
4      if(a < b){
5          System.out.println("a é menor que b");
6      }else{
7          System.out.println("b é menor que a");
8      }
```



## Estrutura condicional ou de seleção - SWITCH CASE

A estrutura de seleção "switch case" em Java é uma alternativa ao "if-else" para verificar várias condições de igualdade. É usada quando há vários valores possíveis para uma variável e cada valor requer uma ação diferente.

A estrutura "switch case" funciona da seguinte maneira: você fornece um valor a ser comparado (geralmente uma variável) e, em seguida, lista vários casos possíveis (identificados pelas palavras-chave "case"). Cada caso é seguido por uma ação a ser executada se o valor for igual ao caso especificado. O comando "break" é usado para interromper a execução da estrutura assim que uma correspondência for encontrada. Se não houver correspondência, o comando "default" é executado.

Exemplo de uso:

```
1 int diaSemana = 2;
2 switch (diaSemana) {
3     case 1:
4         System.out.println("Domingo");
5         break;
6     case 2:
7         System.out.println("Segunda-feira");
8         break;
9     case 3:
10        System.out.println("Terça-feira");
11        break;
12    default:
13        System.out.println("Outro dia da semana");
14 }
```



## Estrutura de repetição - WHILE

A estrutura de repetição "while" em Java é uma estrutura de loop que permite que um trecho de código seja repetido enquanto uma determinada condição for verdadeira. A condição é verificada no início de cada iteração do loop, e se a condição for falsa, a execução do loop é interrompida.

Exemplo de uso:

```
1 int contador = 0;
2 while (contador < 5) {
3     System.out.println("Contador: " + contador);
4     contador = contador + 1;
5 }
```

## Estrutura de repetição - DO WHILE

A estrutura de repetição "do while" em Java é semelhante à estrutura "while", mas com uma diferença importante: o código dentro do loop é executado pelo menos uma vez, independentemente da condição ser verdadeira ou falsa. A condição é verificada no final de cada iteração do loop, e se a condição for falsa, a execução do loop é interrompida.

Exemplo de uso:

```
1 int contador = 0;
2 do {
3     System.out.println("Contador: " + contador);
4     contador = contador + 1;
5 } while (contador < 5);
```



## Estrutura de repetição – FOR

A estrutura de repetição "for" em Java é uma estrutura de loop que permite que um trecho de código seja repetido um determinado número de vezes. Ela é usada quando você sabe quantas vezes um loop deve ser executado.

Exemplo de uso:

```
1 for (int contador = 0; contador < 5; contador++) {  
2     System.out.println("Contador: " + contador);  
3 }
```

## Função

Em Java, uma função é um trecho de código que realiza uma tarefa específica e pode ser chamada a partir de outras partes do programa. Uma função é definida com um nome e pode receber argumentos (ou parâmetros) e retornar um valor.

Uma função tem a seguinte syntax:

```
1 tipo_de_retorno nome_da_função(tipo_de_parâmetro nome_do_parâmetro, ...) {  
2     // corpo da função  
3     return valor_de_retorno;  
4 }
```

O tipo de retorno é o tipo do valor retornado pela função (por exemplo, int, float, String, etc.). Se a função não retorna nenhum valor, o tipo de retorno é void.



Exemplo de uso:

```
1 int soma(int x, int y) {  
2     return x + y;  
3 }
```

Neste exemplo, a função "soma" recebe dois parâmetros inteiros, x e y, e retorna a soma desses valores. A função pode ser chamada a partir de outras partes do programa, passando valores para x e y como argumentos.

Por exemplo:

```
1 int resultado = soma(3, 4); // resultado = 7
```

## Vetores

Em Java, os vetores são objetos que representam arrays de dados de um tipo específico. Eles são declarados com a seguinte sintaxe:

```
1 <tipo>[] <nome do vetor> = new <tipo>[<tamanho do vetor>];
```



Exemplo de uso:



```
1 int[] numbers = new int[10];
```

Isso cria um vetor de inteiros com 10 posições, onde cada posição pode armazenar um valor inteiro. Os valores nas posições de um vetor podem ser acessados e modificados usando a seguinte sintaxe:



```
1 <nome do vetor>[<índice>]= <valor>;
```

Exemplo de uso:



```
1 numbers[0]= 42;
```

Isso atribui o valor 42 à primeira posição do vetor "numbers".

Além disso, é importante lembrar que os índices de um vetor em Java começam em 0 e terminam em tamanho do vetor - 1.



## Exercícios de estrutura sequencial

- <1>. Faça um Programa que mostre a mensagem "Alo mundo" na tela.
- <2>. Faça um Programa que peça um número e então mostre a mensagem O número informado foi [número].
- <3>. Faça um Programa que peça dois números e imprima a soma.
- <4>. Faça um Programa que peça as 4 notas bimestrais e mostre a média.
- <5>. Faça um Programa que converta metros para centímetros.
- <6>. Faça um Programa que peça o raio de um círculo, calcule e mostre sua área.
- <7>. Faça um Programa que calcule a área de um quadrado, em seguida mostre o dobro desta área para o usuário.
- <8>. Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês.
- <9>. Faça um Programa que peça a temperatura em graus Fahrenheit, transforme e mostre a temperatura em graus Celsius. Fórmula =>  $C = 5 * ((F-32) / 9)$ .
- <10>. Faça um Programa que peça a temperatura em graus Celsius, transforme e mostre em graus Fahrenheit.
- <11>. Faça um Programa que peça 2 números inteiros e um número real. Calcule e mostre:
  - a. o produto do dobro do primeiro com metade do segundo.
  - b. a soma do triplo do primeiro com o terceiro.
  - c. o terceiro elevado ao cubo.
- <12>. Tendo como dados de entrada a altura de uma pessoa, construa um algoritmo que calcule seu peso ideal, usando a seguinte fórmula:  $(72.7 * altura) - 58$





- <13>. Tendo como dado de entrada a altura (h) de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:
- Para homens:  $(72.7 * h) - 58$
  - Para mulheres:  $(62.1 * h) - 44.7$

<14>. João Papo-de-Pescador, homem de bem, comprou um microcomputador para controlar o rendimento diário de seu trabalho. Toda vez que ele traz um peso de peixes maior que o estabelecido pelo regulamento de pesca do estado de São Paulo (50 quilos) deve pagar uma multa de R\$ 4,00 por quilo excedente. João precisa que você faça um programa que leia a variável peso (peso de peixes) e calcule o excesso. Gravar na variável "excesso" a quantidade de quilos além do limite e na variável multa o valor da multa que João deverá pagar. Imprima os dados do programa com as mensagens adequadas.

- <15>. Faça um Programa que pergunte quanto você ganha por hora e o número de horas trabalhadas no mês. Calcule e mostre o total do seu salário no referido mês, sabendo-se que são descontados 11% para o Imposto de Renda, 8% para o INSS e 5% para o sindicato, faça um programa que nos dê:
- Salário bruto.
  - Quanto pagou ao INSS.
  - Quanto pagou ao sindicato.
  - O salário líquido.

calcule os descontos e o salário líquido, conforme a tabela abaixo:

- |      |                   |       |
|------|-------------------|-------|
| i.   | + Salário Bruto   | : R\$ |
| ii.  | - IR (11%)        | : R\$ |
| iii. | - INSS (8%)       | : R\$ |
| iv.  | - Sindicato ( 5%) | : R\$ |
| v.   | = Salário Líquido | : R\$ |

Obs.: Salário Bruto - Descontos = Salário Líquido.

- <16>. Faça um programa para uma loja de tintas. O programa deverá pedir o tamanho em metros quadrados da área a ser pintada. Considere que a cobertura da tinta é de 1 litro para cada 3 metros quadrados e que a tinta é vendida em latas de 18 litros, que custam R\$ 80,00. Informe ao usuário a quantidades de latas de tinta a serem compradas e o preço total.



<17>. Faça um programa que peça o tamanho de um arquivo para download (em MB) e a velocidade de um link de Internet (em Mbps), calcule e informe o tempo aproximado de download do arquivo usando este link (em minutos).

## Exercícios de estrutura de decisão

<18>. Faça um Programa que peça dois números e imprima o maior deles.

<19>. Faça um Programa que peça um valor e mostre na tela se o valor é positivo ou negativo.

<20>. Faça um Programa que verifique se uma letra digitada é "F" ou "M". Conforme a letra escrever: F - Feminino, M - Masculino, Sexo Inválido.

<21>. Faça um Programa que verifique se uma letra digitada é vogal ou consoante.

<22>. Faça um programa para a leitura de duas notas parciais de um aluno. O programa deve calcular a média alcançada por aluno e apresentar:

- a. A mensagem "Aprovado", se a média alcançada for maior ou igual a sete;
- b. A mensagem "Reprovado", se a média for menor do que sete;
- c. A mensagem "Aprovado com Distinção", se a média for igual a dez.

<23>. Faça um Programa que leia três números e mostre o maior deles.

<24>. Faça um Programa que leia três números e mostre o maior e o menor deles.

<25>. Faça um programa que pergunte o preço de três produtos e informe qual produto você deve comprar, sabendo que a decisão é sempre pelo mais barato.

<26>. Faça um Programa que leia três números e mostre-os em ordem decrescente.

<27>. Faça um Programa que pergunte em que turno você estuda. Peça para digitar M-matutino ou V-Vespertino ou N- Noturno. Imprima a mensagem "Bom Dia!", "Boa Tarde!" ou "Boa Noite!" ou "Valor Inválido!", conforme o caso.



<28>. As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores e lhe contraram para desenvolver o programa que calculará os reajustes. Faça um programa que recebe o salário de um colaborador e o reajuste segundo o seguinte critério, baseado no salário atual:

- a. Salários até R\$ 280,00 (incluindo): aumento de 20%
- b. Salários entre R\$ 280,00 e R\$ 700,00: aumento de 15%
- c. Salários entre R\$ 700,00 e R\$ 1500,00: aumento de 10%
- d. Salários de R\$ 1500,00 em diante: aumento de 5%

Após o aumento ser realizado, informe na tela:

- e. O salário antes do reajuste;
- f. O percentual de aumento aplicado;
- g. O valor do aumento;
- h. O novo salário, após o aumento.

<29>. Faça um programa para o cálculo de uma folha de pagamento, sabendo que os descontos são do Imposto de Renda, que depende do salário bruto (conforme tabela abaixo) e 3% para o Sindicato e que o FGTS corresponde a 11% do Salário Bruto, mas não é descontado (é a empresa que deposita). O Salário Líquido corresponde ao Salário Bruto menos os descontos. O programa deverá pedir ao usuário o valor da sua hora e a quantidade de horas trabalhadas no mês. Desconto do IR:

- a. Salário Bruto até 900 (inclusive) - isento
- b. Salário Bruto até 1500 (inclusive) - desconto de 5%
- c. Salário Bruto até 2500 (inclusive) - desconto de 10%
- d. Salário Bruto acima de 2500 - desconto de 20% Imprima na tela as informações, dispostas conforme o exemplo abaixo. No exemplo o valor da hora é 5 e a quantidade de hora é 220.

i.	Salário Bruto: (5 * 220)	: R\$ 1100,00
ii.	(-) IR (5%)	: R\$ 55,00
iii.	(-) INSS ( 10%)	: R\$ 110,00
iv.	FGTS (11%)	: R\$ 121,00
v.	Total de descontos	: R\$ 165,00
vi.	Salário Líquido	: R\$ 935,00



<30>. Faça um programa para o cálculo de uma folha de pagamento, sabendo que os descontos são do Imposto de Renda, que depende do salário bruto (conforme tabela abaixo) e 3% para o Sindicato e que o FGTS corresponde a 11% do Salário Bruto, mas não é descontado (é a empresa que deposita). O Salário Líquido corresponde ao Salário Bruto menos os descontos. O programa deverá pedir ao usuário o valor da sua hora e a quantidade de horas trabalhadas no mês.

Desconto do IR:

- a. Salário Bruto até 900 (inclusive) - isento
- b. Salário Bruto até 1500 (inclusive) - desconto de 5%
- c. Salário Bruto até 2500 (inclusive) - desconto de 10%
- d. Salário Bruto acima de 2500 - desconto de 20% Imprima na tela as informações, dispostas conforme o exemplo abaixo. No exemplo o valor da hora é 5 e a quantidade de hora é 220.

i.	Salário Bruto: (5 * 220)	: R\$ 1100,00
ii.	(-) IR (5%)	: R\$ 55,00
iii.	(-) INSS ( 10%)	: R\$ 110,00
iv.	FGTS (11%)	: R\$ 121,00
v.	Total de descontos	: R\$ 165,00
vi.	Salário Líquido	: R\$ 935,00

<31>. Faça um Programa que leia um número e exiba o dia correspondente da semana. (1-Domingo, 2- Segunda, etc.), se digitar outro valor deve aparecer valor inválido.

<32>. Faça um programa que lê as duas notas parciais obtidas por um aluno numa disciplina ao longo de um semestre, e calcule a sua média. A atribuição de conceitos obedece à tabela abaixo:

i.	Média de Aproveitamento	Conceito
ii.	Entre 9.0 e 10.0	A
iii.	Entre 7.5 e 9.0	B
iv.	Entre 6.0 e 7.5	C
v.	Entre 4.0 e 6.0	D
vi.	Entre 4.0 e zero	E

O algoritmo deve mostrar na tela as notas, a média, o conceito correspondente e a mensagem “APROVADO” se o conceito for A, B ou C ou “REPROVADO” se o conceito for D ou E.



<33>. Faça um Programa que peça os 3 lados de um triângulo. O programa deverá informar se os valores podem ser um triângulo. Indique, caso os lados formem um triângulo, se o mesmo é: equilátero, isósceles ou escaleno.

Dicas:

- a. Três lados formam um triângulo quando a soma de quaisquer dois lados for maior que o terceiro;
- b. Triângulo Equilátero: três lados iguais;
- c. Triângulo Isósceles: quaisquer dois lados iguais;
- d. Triângulo Escaleno: três lados diferentes;

<34>. Faça um programa que calcule as raízes de uma equação do segundo grau, na forma  $ax^2 + bx + c$ . O programa deverá pedir os valores de a, b e c e fazer as consistências, informando ao usuário nas seguintes situações:

- a. Se o usuário informar o valor de A igual a zero, a equação não é do segundo grau e o programa não deve fazer pedir os demais valores, sendo encerrado;
- b. Se o delta calculado for negativo, a equação não possui raízes reais. Informe ao usuário e encerre o programa;
- c. Se o delta calculado for igual a zero a equação possui apenas uma raiz real; informe-a ao usuário;
- d. Se o delta for positivo, a equação possui duas raiz reais; informe-as ao usuário;

<35>. Faça um Programa que peça um número correspondente a um determinado ano e em seguida informe se este ano é ou não bissexto.

<36>. Faça um Programa que peça uma data no formato dd/mm/aaaa e determine se a mesma é uma data válida.

<37>. Faça um Programa que leia um número inteiro menor que 1000 e imprima a quantidade de centenas, dezenas e unidades do mesmo.

Observando os termos no plural a colocação do "e", da vírgula entre outros.

Exemplo:

- a. 326 = 3 centenas, 2 dezenas e 6 unidades
- b. 12 = 1 dezena e 2 unidades Testar com: 326, 300, 100, 320, 310, 305, 301, 101, 311, 111, 25, 20, 10, 21, 11, 1, 7 e 16



- <38>. Faça um Programa para leitura de três notas parciais de um aluno. O programa deve calcular a média alcançada por aluno e apresentar:
- A mensagem "Aprovado", se a média for maior ou igual a 7, com a respectiva média alcançada;
  - A mensagem "Reprovado", se a média for menor do que 7, com a respectiva média alcançada;
  - A mensagem "Aprovado com Distinção", se a média for igual a 10.
- <39>. Faça um Programa para um caixa eletrônico. O programa deverá perguntar ao usuário a valor do saque e depois informar quantas notas de cada valor serão fornecidas. As notas disponíveis serão as de 1, 5, 10, 50 e 100 reais. O valor mínimo é de 10 reais e o máximo de 600 reais. O programa não deve se preocupar com a quantidade de notas existentes na máquina.
- Exemplo 1: Para sacar a quantia de 256 reais, o programa fornece duas notas de 100, uma nota de 50, uma nota de 5 e uma nota de 1;
  - Exemplo 2: Para sacar a quantia de 399 reais, o programa fornece três notas de 100, uma nota de 50, quatro notas de 10, uma nota de 5 e quatro notas de 1.
- <40>. Faça um Programa que peça um número inteiro e determine se ele é par ou ímpar. Dica: utilize o operador módulo (resto da divisão).
- <41>. Faça um Programa que peça um número e informe se o número é inteiro ou decimal. Dica: utilize uma função de arredondamento.
- <42>. Faça um Programa que leia 2 números e em seguida pergunte ao usuário qual operação ele deseja realizar. O resultado da operação deve ser acompanhado de uma frase que diga se o número é:
- par ou ímpar;
  - positivo ou negativo;
  - inteiro ou decimal.
- <43>. Faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:
- "Telefonou para a vítima?"
  - "Esteve no local do crime?"
  - "Mora perto da vítima?"
  - "Devia para a vítima?"
  - "Já trabalhou com a vítima?"
- O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".



<44>. Um posto está vendendo combustíveis com a seguinte tabela de descontos:

a. Álcool:

i. até 20 litros, desconto de 3% por litro

ii. acima de 20 litros, desconto de 5% por litro

b. Gasolina:

i. até 20 litros, desconto de 4% por litro

ii. acima de 20 litros, desconto de 6% por litro

Escreva um algoritmo que leia o número de litros vendidos, o tipo de combustível (codificado da seguinte forma: A-álcool, G-gasolina), calcule e imprima o valor a ser pago pelo cliente sabendo-se que o preço do litro da gasolina é R\$ 2,50 o preço do litro do álcool é R\$ 1,90.

<45>. Uma fruteira está vendendo frutas com a seguinte tabela de preços:

	Até 5 Kg	Acima de 5 Kg
Morango	R\$ 2,50 por Kg	R\$ 2,20 por Kg
Maçã	R\$ 1,80 por Kg	R\$ 1,50 por Kg

Se o cliente comprar mais de 8 Kg em frutas ou o valor total da compra ultrapassar R\$ 25,00, receberá ainda um desconto de 10% sobre este total. Escreva um algoritmo para ler a quantidade (em Kg) de morangos e a quantidade (em Kg) de maçãs adquiridas e escreva o valor a ser pago pelo cliente.

<46>. O Hipermercado Tabajara está com uma promoção de carnes que é imperdível. Confira:

	Até 5 Kg	Acima de 5 Kg
File Duplo	R\$ 4,90 por Kg	R\$ 5,80 por Kg
Alcatra	R\$ 5,90 por Kg	R\$ 6,80 por Kg
Picanha	R\$ 6,90 por Kg	R\$ 7,80 por Kg

Para atender a todos os clientes, cada cliente poderá levar apenas um dos tipos de carne da promoção, porém não há limites para a quantidade de carne por cliente. Se compra for feita no cartão Tabajara o cliente receberá ainda um desconto de 5% sobre o total da compra. Escreva um programa que peça o tipo e a quantidade de carne comprada pelo usuário e gere um cupom fiscal, contendo as informações da compra: tipo e quantidade de carne, preço total, tipo de pagamento, valor do desconto e valor a pagar.



## Exercícios de estrutura de repetição

- <47>. Faça um programa que peça uma nota, entre zero e dez. Mostre uma mensagem caso o valor seja inválido e continue pedindo até que o usuário informe um valor válido.
- <48>. Faça um programa que leia um nome de usuário e a sua senha e não aceite a senha igual ao nome do usuário, mostrando uma mensagem de erro e voltando a pedir as informações.
- <49>. Faça um programa que leia e valide as seguintes informações:
- a. Nome: maior que 3 caracteres;
  - b. Idade: entre 0 e 150;
  - c. Salário: maior que zero;
  - d. Sexo: 'f' ou 'm';
  - e. Estado Civil: 's', 'c', 'v', 'd';
- <50>. Supondo que a população de um país A seja da ordem de 80000 habitantes com uma taxa anual de crescimento de 3% e que a população de B seja 200000 habitantes com uma taxa de crescimento de 1.5%. Faça um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.
- <51>. Altere o programa anterior permitindo ao usuário informar as populações e as taxas de crescimento iniciais. Valide a entrada e permita repetir a operação.
- <52>. Faça um programa que imprima na tela os números de 1 a 20, um abaixo do outro. Depois modifique o programa para que ele mostre os números um ao lado do outro.
- <53>. Faça um programa que leia 5 números e informe o maior número.
- <54>. Faça um programa que leia 5 números e informe a soma e a média dos números.
- <55>. Faça um programa que imprima na tela apenas os números ímpares entre 1 e 50.
- <56>. Faça um programa que receba dois números inteiros e gere os números inteiros que estão no intervalo compreendido por eles.
- <57>. Altere o programa anterior para mostrar no final a soma dos números.





- <58>. Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual número ele deseja ver a tabuada. A saída deve ser conforme o exemplo abaixo:
- i. Tabuada de 5:
  - ii.  $5 \times 1 = 5$
  - iii.  $5 \times 2 = 10$
  - iv. ...
  - v.  $5 \times 10 = 50$
- <59>. Faça um programa que peça dois números, base e expoente, calcule e mostre o primeiro número elevado ao segundo número. Não utilize a função de potência da linguagem.
- <60>. Faça um programa que peça 10 números inteiros, calcule e mostre a quantidade de números pares e a quantidade de números ímpares.
- <61>. A série de Fibonacci é formada pela seqüência 1,1,2,3,5,8,13,21,34,55,... Faça um programa capaz de gerar a série até o n-ésimo termo.
- <62>. A série de Fibonacci é formada pela seqüência 0,1,1,2,3,5,8,13,21,34,55,... Faça um programa que gere a série até que o valor seja maior que 500.
- <63>. Faça um programa que calcule o fatorial de um número inteiro fornecido pelo usuário. Ex.:  $5! = 5.4.3.2.1 = 120$
- <64>. Faça um programa que, dado um conjunto de N números, determine o menor valor, o maior valor e a soma dos valores.
- <65>. Altere o programa anterior para que ele aceite apenas números entre 0 e 1000.
- <66>. Altere o programa de cálculo do fatorial, permitindo ao usuário calcular o fatorial várias vezes e limitando o fatorial a números inteiros positivos e menores que 16.
- <67>. Faça um programa que peça um número inteiro e determine se ele é ou não um número primo. Um número primo é aquele que é divisível somente por ele mesmo e por 1.
- <68>. Altere o programa de cálculo dos números primos, informando, caso o número não seja primo, por quais número ele é divisível.



- <69>. Faça um programa que mostre todos os primos entre 1 e N sendo N um número inteiro fornecido pelo usuário. O programa deverá mostrar também o número de divisões que ele executou para encontrar os números primos. Serão avaliados o funcionamento, o estilo e o número de testes (divisões) executados.
- <70>. Faça um programa que calcule o mostre a média aritmética de N notas.
- <71>. Faça um programa que peça para n pessoas a sua idade, ao final o programa devera verificar se a média de idade da turma varia entre 0 e 25,26 e 60 e maior que 60; e então, dizer se a turma é jovem, adulta ou idosa, conforme a média calculada.
- <72>. Numa eleição existem três candidatos. Faça um programa que peça o número total de eleitores. Peça para cada eleitor votar e ao final mostrar o número de votos de cada candidato.
- <73>. Faça um programa que calcule o número médio de alunos por turma. Para isto, peça a quantidade de turmas e a quantidade de alunos para cada turma. As turmas não podem ter mais de 40 alunos.
- <74>. Faça um programa que calcule o valor total investido por um colecionador em sua coleção de CDs e o valor médio gasto em cada um deles. O usuário deverá informar a quantidade de CDs e o valor para em cada um.
- <75>. O Sr. Manoel Joaquim possui uma grande loja de artigos de R\$ 1,99, com cerca de 10 caixas. Para agilizar o cálculo de quanto cada cliente deve pagar ele desenvolveu um tabela que contém o número de itens que o cliente comprou e ao lado o valor da conta. Desta forma a atendente do caixa precisa apenas contar quantos itens o cliente está levando e olhar na tabela de preços. Você foi contratado para desenvolver o programa que monta esta tabela de preços, que conterá os preços de 1 até 50 produtos, conforme o exemplo abaixo:
- i. Lojas Quase Dois - Tabela de preços
  - ii. 1 - R\$ 1.99
  - iii. 2 - R\$ 3.98
  - iv. ...
  - v. 50 - R\$ 99.50



<76>. O Sr. Manoel Joaquim acaba de adquirir uma panificadora e pretende implantar a metodologia da tabelinha, que já é um sucesso na sua loja de 1,99. Você foi contratado para desenvolver o programa que monta a tabela de preços de pães, de 1 até 50 pães, a partir do preço do pão informado pelo usuário, conforme o exemplo abaixo:

- i. Preço do pão: R\$ 0.18
- ii. Panificadora Pão de Ontem - Tabela de preços
- iii. 1 - R\$ 0.18
- iv. 2 - R\$ 0.36
- v. ...
- vi. 50 - R\$ 9.00

<77>. O Sr. Manoel Joaquim expandiu seus negócios para além dos negócios de 1,99 e agora possui uma loja de conveniências. Faça um programa que implemente uma caixa registradora rudimentar. O programa deverá receber um número desconhecido de valores referentes aos preços das mercadorias. Um valor zero deve ser informado pelo operador para indicar o final da compra. O programa deve então mostrar o total da compra e perguntar o valor em dinheiro que o cliente forneceu, para então calcular e mostrar o valor do troco. Após esta operação, o programa deverá voltar ao ponto inicial, para registrar a próxima compra. A saída deve ser conforme o exemplo abaixo:

- i. Lojas Tabajara
- ii. Produto 1: R\$ 2.20
- iii. Produto 2: R\$ 5.80
- iv. Produto 3: R\$ 0
- v. Total: R\$ 9.00
- vi. Dinheiro: R\$ 20.00
- vii. Troco: R\$ 11.00
- viii. ...

<78>. Faça um programa que calcule o fatorial de um número inteiro fornecido pelo usuário. Ex.:  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$ . A saída deve ser conforme o exemplo abaixo:

- i. Fatorial de: 5
- ii.  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

<79>. O Departamento Estadual de Meteorologia lhe contratou para desenvolver um programa que leia as um conjunto indeterminado de temperaturas, e informe ao final a menor e a maior temperaturas informadas, bem como a média das temperaturas.



- <80>. Os números primos possuem várias aplicações dentro da Computação, por exemplo na Criptografia. Um número primo é aquele que é divisível apenas por um e por ele mesmo. Faça um programa que peça um número inteiro e determine se ele é ou não um número primo.
- <81>. Encontrar números primos é uma tarefa difícil. Faça um programa que gera uma lista dos números primos existentes entre 1 e um número inteiro informado pelo usuário.
- <82>. Desenvolva um programa que faça a tabuada de um número qualquer inteiro que será digitado pelo usuário, mas a tabuada não deve necessariamente iniciar em 1 e terminar em 10, o valor inicial e final devem ser informados também pelo usuário, conforme exemplo abaixo:
- a. Montar a tabuada de: 5
    - i. Começar por: 4
    - ii. Terminar em: 7
  - b. Vou montar a tabuada de 5 começando em 4 e terminando em 7:
    - i.  $5 \times 4 = 20$
    - ii.  $5 \times 5 = 25$
    - iii.  $5 \times 6 = 30$
    - iv.  $5 \times 7 = 35$
- Obs: Você deve verificar se o usuário não digitou o final menor que o inicial.



<83>. Uma academia deseja fazer um senso entre seus clientes para descobrir o mais alto, o mais baixo, a mais gordo e o mais magro, para isto você deve fazer um programa que pergunte a cada um dos clientes da academia seu código, sua altura e seu peso. O final da digitação de dados deve ser dada quando o usuário digitar 0 (zero) no campo código. Ao encerrar o programa também devem ser informados os códigos e valores do cliente mais alto, do mais baixo, do mais gordo e do mais magro, além da média das alturas e dos pesos dos clientes.

- a. Um funcionário de uma empresa recebe aumento salarial anualmente: Sabe-se que:
- b. Esse funcionário foi contratado em 1995, com salário inicial de R\$ 1.000,00;
- c. Em 1996 recebeu aumento de 1,5% sobre seu salário inicial;
- d. A partir de 1997 (inclusive), os aumentos salariais sempre correspondem ao dobro do percentual do ano anterior.

Faça um programa que determine o salário atual desse funcionário. Após concluir isto, altere o programa permitindo que o usuário digite o salário inicial do funcionário.

<84>. Faça um programa que leia dez conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando a sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Mostre o número do aluno mais alto e o número do aluno mais baixo, junto com suas alturas.

<85>. Foi feita uma estatística em cinco cidades brasileiras para coletar dados sobre acidentes de trânsito. Foram obtidos os seguintes dados:

- a. Código da cidade;
- b. Número de veículos de passeio (em 1999);
- c. Número de acidentes de trânsito com vítimas (em 1999). Deseja-se saber:
- d. Qual o maior e menor índice de acidentes de transito e a que cidade pertence;
- e. Qual a média de veículos nas cinco cidades juntas;
- f. Qual a média de acidentes de trânsito nas cidades com menos de 2.000 veículos de passeio.



<86>. Faça um programa que receba o valor de uma dívida e mostre uma tabela com os seguintes dados: valor da dívida, valor dos juros, quantidade de parcelas e valor da parcela.

Os juros e a quantidade de parcelas seguem a tabela abaixo:

Quantidade de Parcelas % de Juros sobre o valor inicial da dívida

1	0
3	10
6	15
9	20
12	25

Exemplo de saída do programa:

Valor da Dívida Valor dos Juros Quantidade de Parcelas Valor da Parcela

R\$ 1.000,00	0	1	R\$ 1.000,00
R\$ 1.100,00	100	3	R\$ 366,00
R\$ 1.150,00	150	6	R\$ 191,67

<87>. O cardápio de uma lanchonete é o seguinte:

Especificação	Código	Preço
Cachorro Quente	100	R\$ 1,20
Bauru Simples	101	R\$ 1,30
Bauru com ovo	102	R\$ 1,50
Hambúrguer	103	R\$ 1,20
Cheeseburger	104	R\$ 1,30
Refrigerante	105	R\$ 1,00

Faça um programa que leia o código dos itens pedidos e as quantidades desejadas. Calcule e mostre o valor a ser pago por item (preço \* quantidade) e o total geral do pedido. Considere que o cliente deve informar quando o pedido deve ser encerrado.



<88>. Em uma eleição presidencial existem quatro candidatos. Os votos são informados por meio de código. Os códigos utilizados são:

1 , 2, 3, 4 - Votos para os respectivos candidatos  
(você deve montar a tabela ex: 1 - Jose/ 2- João/etc)  
5 - Voto Nulo  
6 - Voto em Branco

Faça um programa que calcule e mostre:

- a. O total de votos para cada candidato;
- b. O total de votos nulos;
- c. O total de votos em branco;
- d. A percentagem de votos nulos sobre o total de votos;
- e. A percentagem de votos em branco sobre o total de votos. Para finalizar o conjunto de votos tem-se o valor zero.

<89>. Desenvolver um programa para verificar a nota do aluno em uma prova com 10 questões, o programa deve perguntar ao aluno a resposta de cada questão e ao final comparar com o gabarito da prova e assim calcular o total de acertos e a nota (atribuir 1 ponto por resposta certa). Após cada aluno utilizar o sistema deve ser feita uma pergunta se outro aluno vai utilizar o sistema. Após todos os alunos terem respondido informar:

- a. Maior e Menor Acerto;
- b. Total de Alunos que utilizaram o sistema;
- c. A Média das Notas da Turma.

Gabarito da Prova:

01 - A  
02 - B  
03 - C  
04 - D  
05 - E  
06 - E  
07 - D  
08 - C  
09 - B  
10 - A

Após concluir isto você poderia incrementar o programa permitindo que o professor digite o gabarito da prova antes dos alunos usarem o programa.



<90>. Em uma competição de salto em distância cada atleta tem direito a cinco saltos. No final da série de saltos de cada atleta, o melhor e o pior resultados são eliminados. O seu resultado fica sendo a média dos três valores restantes. Você deve fazer um programa que receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos e depois informe a média dos saltos conforme a descrição acima informada (retirar o melhor e o pior salto e depois calcular a média). Faça uso de uma lista para armazenar os saltos. Os saltos são informados na ordem da execução, portanto não são ordenados. O programa deve ser encerrado quando não for informado o nome do atleta. A saída do programa deve ser conforme o exemplo abaixo:

Atleta: Rodrigo Curvêllo

Primeiro Salto: 6.5 m

Segundo Salto: 6.1 m

Terceiro Salto: 6.2 m

Quarto Salto: 5.4 m

Quinto Salto: 5.3 m

Melhor salto: 6.5 m

Pior salto: 5.3 m

Média dos demais saltos: 5.9 m

Resultado final:

Rodrigo Curvêllo: 5.9 m





<91>. Em uma competição de ginástica, cada atleta recebe votos de sete jurados. A melhor e a pior nota são eliminadas. A sua nota fica sendo a média dos votos restantes. Você deve fazer um programa que receba o nome do ginasta e as notas dos sete jurados alcançadas pelo atleta em sua apresentação e depois informe a sua média, conforme a descrição acima informada (retirar o melhor e o pior salto e depois calcular a média com as notas restantes). As notas não são informados ordenadas. Um exemplo de saída do programa deve ser conforme o exemplo abaixo:

```
Atleta: Aparecido Parente
Nota: 9.9
Nota: 7.5
Nota: 9.5
Nota: 8.5
Nota: 9.0
Nota: 8.5
Nota: 9.7
```

```
Resultado final:
Atleta: Aparecido Parente
Melhor nota: 9.9
Pior nota: 7.5
Média: 9,04
```

<92>. Faça um programa que peça um número inteiro positivo e em seguida mostre este número invertido.

Exemplo:

```
12376489
=> 98467321
```

<93>. Faça um programa que mostre os n termos da Série a seguir:

$$S = 1/1 + 2/3 + 3/5 + 4/7 + 5/9 + \dots + n/m.$$

Imprima no final a soma da série.

<94>. Sendo  $H = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ , Faça um programa que calcule o valor de H com N termos.

<95>. Faça um programa que mostre os n termos da Série a seguir:

$$S = 1/1 + 2/3 + 3/5 + 4/7 + 5/9 + \dots + n/m.$$

Imprima no final a soma da série.



## Exercício com arquivos

<96>. Faça um programa que leia um arquivo texto contendo uma lista de endereços IP e gere um outro arquivo, contendo um relatório dos endereços IP válidos e inválidos. O arquivo de entrada possui o seguinte formato:

```
200.135.80.9
192.168.1.1
8.35.67.74
257.32.4.5
85.345.1.2
1.2.3.4
9.8.234.5
192.168.0.256
```

O arquivo de saída possui o seguinte formato:

```
[Endereços válidos:]
200.135.80.9
192.168.1.1
8.35.67.74
1.2.3.4

[Endereços inválidos:]
257.32.4.5
85.345.1.2
9.8.234.5
192.168.0.256
```

## Exercícios com strings

<97>. Tamanho de strings. Faça um programa que leia 2 strings e informe o conteúdo delas seguido do seu comprimento. Informe também se as duas strings possuem o mesmo comprimento e são iguais ou diferentes no conteúdo.

```
Compara duas strings
String 1: Brasil Hexa 2006
String 2: Brasil! Hexa 2006!
Tamanho de "Brasil Hexa 2006": 16 caracteres
Tamanho de "Brasil! Hexa 2006!": 18 caracteres
As duas strings são de tamanhos diferentes.
```



As duas strings possuem conteúdo diferente.

<98>. Nome ao contrário em maiúsculas. Faça um programa que permita ao usuário digitar o seu nome e em seguida mostre o nome do usuário de trás para frente utilizando somente letras maiúsculas. Dica: lembre-se que ao informar o nome o usuário pode digitar letras maiúsculas ou minúsculas.

<99>. Nome na vertical. Faça um programa que solicite o nome do usuário e imprima-o na vertical.

```
F
U
L
A
N
O
```

<100>. Nome na vertical em escada. Modifique o programa anterior de forma a mostrar o nome em formato de escada.

```
F
FU
FUL
FULA
FULAN
FULANO
```

<101>. Nome na vertical em escada invertida. Altere o programa anterior de modo que a escada seja invertida.

```
FULANO
FULAN
FULA
FUL
FU
F
```

<102>. Data por extenso. Faça um programa que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso.

```
Data de Nascimento: 29/10/1973
Você nasceu em  29 de Outubro de 1973.
```



- <103>. Conta espaços e vogais. Dado uma string com uma frase informada pelo usuário (incluindo espaços em branco), conte:
- a. quantos espaços em branco existem na frase.
  - b. quantas vezes aparecem as vogais a, e, i, o, u.
- <104>. Palíndromo. Um palíndromo é uma seqüência de caracteres cuja leitura é idêntica se feita da direita para esquerda ou vice-versa. Por exemplo: OSSO e OVO são palíndromos. Em textos mais complexos os espaços e pontuação são ignorados. A frase SUBI NO ONIBUS é o exemplo de uma frase palíndroma onde os espaços foram ignorados. Faça um programa que leia uma seqüência de caracteres, mostre-a e diga se é um palíndromo ou não.
- <105>. Verificação de CPF. Desenvolva um programa que solicite a digitação de um número de CPF no formato xxx.xxx.xxx-xx e indique se é um número válido ou inválido através da validação dos dígitos verificadores edos caracteres de formatação.
- <106>. Número por extenso. Escreva um programa que solicite ao usuário a digitação de um número até 99 e imprima-o na tela por extenso.
- <107>. Jogo de Forca. Desenvolva um jogo da forca. O programa terá uma lista de palavras lidas de um arquivo texto e escolherá uma aleatoriamente. O jogador poderá errar 6 vezes antes de ser enforcado.

Digite uma letra: A

-> Você errou pela 1ª vez. Tente de novo!

Digite uma letra: O

A palavra é: \_ \_ \_ \_ O

Digite uma letra: E

A palavra é: \_ E \_ \_ O

Digite uma letra: S

-> Você errou pela 2ª vez. Tente de novo!



<108>. Valida e corrige número de telefone. Faça um programa que leia um número de telefone, e corrija o número no caso deste conter somente 7 dígitos, acrescentando o '3' na frente. O usuário pode informar o número com ou sem o traço separador.

```
Valida e corrige número de telefone
```

```
Telefone: 461-0133
```

```
Telefone possui 7 dígitos. Vou acrescentar o digito três na frente.
```

```
Telefone corrigido sem formatação: 34610133
```

```
Telefone corrigido com formatação: 3461-0133
```

<109>. Jogo da palavra embaralhada. Desenvolva um jogo em que o usuário tenha que adivinhar uma palavra que será mostrada com as letras embaralhadas. O programa terá uma lista de palavras lidas de um arquivo texto e escolherá uma aleatoriamente. O jogador terá seis tentativas para adivinhar a palavra. Ao final a palavra deve ser mostrada na tela, informando se o usuário ganhou ou perdeu o jogo.

<110>. Leet spek generator. Leet é uma forma de se escrever o alfabeto latino usando outros símbolos em lugar das letras, como números por exemplo. A própria palavra leet admite muitas variações, como l33t ou 1337. O uso do leet reflete uma subcultura relacionada ao mundo dos jogos de computador e internet, sendo muito usada para confundir os iniciantes e afirmar-se como parte de um grupo. Pesquise sobre as principais formas de traduzir as letras. Depois, faça um programa que peça uma texto e transforme-o para a grafia leet speak.



## Exercícios com listas

- <111>. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.
- <112>. Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.
- <113>. Faça um Programa que leia 4 notas, mostre as notas e a média na tela.
- <114>. Faça um Programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.
- <115>. Faça um Programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números IMPARES no vetor impar. Imprima os três vetores.
- <116>. Faça um Programa que peça as quatro notas de 10 alunos, calcule e armazene num vetor a média de cada aluno, imprima o número de alunos com média maior ou igual a 7.0.
- <117>. Faça um Programa que leia um vetor de 5 números inteiros, mostre a soma, a multiplicação e os números.
- <118>. Faça um Programa que peça a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima a idade e a altura na ordem inversa a ordem lida.
- <119>. Faça um Programa que leia um vetor A com 10 números inteiros, calcule e mostre a soma dos quadrados dos elementos do vetor.
- <120>. Faça um Programa que leia dois vetores com 10 elementos cada. Gere um terceiro vetor de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados dos dois outros vetores.
- <121>. Altere o programa anterior, intercalando 3 vetores de 10 elementos cada.
- <122>. Foram anotadas as idades e alturas de 30 alunos. Faça um Programa que determine quantos alunos com mais de 13 anos possuem altura inferior à média de altura desses alunos.



<123>. Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Após isto, calcule a média anual das temperaturas e mostre todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . . ).

<124>. Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:

- a. "Telefonou para a vítima?"
- b. "Esteve no local do crime?"
- c. "Mora perto da vítima?"
- d. "Devia para a vítima?"
- e. "Já trabalhou com a vítima?"

O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".

<125>. Faça um programa que leia um número indeterminado de valores, correspondentes a notas, encerrando a entrada de dados quando for informado um valor igual a -1 (que não deve ser armazenado). Após esta entrada de dados, faça:

- a. Mostre a quantidade de valores que foram lidos;
- b. Exiba todos os valores na ordem em que foram informados, um ao lado do outro;
- c. Exiba todos os valores na ordem inversa à que foram informados, um abaixo do outro;
- d. Calcule e mostre a soma dos valores;
- e. Calcule e mostre a média dos valores;
- f. Calcule e mostre a quantidade de valores acima da média calculada;
- g. Calcule e mostre a quantidade de valores abaixo de sete;
- h. Encerre o programa com uma mensagem;



<126>. Utilize uma lista para resolver o problema a seguir. Uma empresa paga seus vendedores com base em comissões. O vendedor recebe \$200 por semana mais 9 por cento de suas vendas brutas daquela semana. Por exemplo, um vendedor que teve vendas brutas de \$3000 em uma semana recebe \$200 mais 9 por cento de \$3000, ou seja, um total de \$470. Escreva um programa (usando um array de contadores) que determine quantos vendedores receberam salários nos seguintes intervalos de valores:

- a. \$200 - \$299
- b. \$300 - \$399
- c. \$400 - \$499
- d. \$500 - \$599
- e. \$600 - \$699
- f. \$700 - \$799
- g. \$800 - \$899
- h. \$900 - \$999
- i. \$1000 em diante

Desafio: Crie uma fórmula para chegar na posição da lista a partir do salário, sem fazer vários ifs aninhados.

<127>. Em uma competição de salto em distância cada atleta tem direito a cinco saltos. O resultado do atleta será determinado pela média dos cinco valores restantes. Você deve fazer um programa que receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos e depois informe o nome, os saltos e a média dos saltos. O programa deve ser encerrado quando não for informado o nome do atleta. A saída do programa deve ser conforme o exemplo abaixo:

Atleta: Rodrigo Curvêllo

Primeiro Salto: 6.5 m

Segundo Salto: 6.1 m

Terceiro Salto: 6.2 m

Quarto Salto: 5.4 m

Quinto Salto: 5.3 m

Resultado final:

Atleta: Rodrigo Curvêllo

Saltos: 6.5 - 6.1 - 6.2 - 5.4 - 5.3

Média dos saltos: 5.9 m





<128>. Uma grande emissora de televisão quer fazer uma enquete entre os seus telespectadores para saber qual o melhor jogador após cada jogo. Para isto, faz-se necessário o desenvolvimento de um programa, que será utilizado pelas telefonistas, para a computação dos votos. Sua equipe foi contratada para desenvolver este programa, utilizando a linguagem de programação C++. Para computar cada voto, a telefonista digitará um número, entre 1 e 23, correspondente ao número da camisa do jogador. Um número de jogador igual zero, indica que a votação foi encerrada. Se um número inválido for digitado, o programa deve ignorá-lo, mostrando uma breve mensagem de aviso, e voltando a pedir outro número. Após o final da votação, o programa deverá exibir:

- a. O total de votos computados;
- b. Os números e respectivos votos de todos os jogadores que receberam votos;
- c. O percentual de votos de cada um destes jogadores;
- d. O número do jogador escolhido como o melhor jogador da partida, juntamente com o número de votos e o percentual de votos dados a ele.

Observe que os votos inválidos e o zero final não devem ser computados como votos. O resultado aparece ordenado pelo número do jogador. O programa deve fazer uso de arrays. O programa deverá executar o cálculo do percentual de cada jogador através de uma função. Esta função receberá dois parâmetros: o número de votos de um jogador e o total de votos. A função calculará o percentual e retornará o valor calculado. Abaixo segue uma tela de exemplo. A disposição das informações deve ser o mais próxima possível ao exemplo. Os dados são fictícios e podem mudar a cada execução do programa. Ao final, o programa deve ainda gravar os dados referentes ao resultado da votação em um arquivo texto no disco, obedecendo a mesma disposição apresentada na tela.

Enquete: Quem foi o melhor jogador?

```
Número do jogador (0=fim): 9
Número do jogador (0=fim): 10
Número do jogador (0=fim): 9
Número do jogador (0=fim): 10
Número do jogador (0=fim): 11
Número do jogador (0=fim): 10
Número do jogador (0=fim): 50
Informe um valor entre 1 e 23 ou 0 para sair!
Número do jogador (0=fim): 9
Número do jogador (0=fim): 9
```



Número do jogador (0=fim): 0

Resultado da votação:

Foram computados 8 votos.

Jogador	Votos	%
9	4	50,0%
10	3	37,5%
11	1	12,5%

O melhor jogador foi o número 9, com 4 votos, correspondendo a 50% do total de votos.

<129>. Uma empresa de pesquisas precisa tabular os resultados da seguinte enquete feita a uma grande quantidade de organizações:

"Qual o melhor Sistema Operacional para uso em servidores?"

As possíveis respostas são:

- 1- Windows Server
- 2- Unix
- 3- Linux
- 4- Netware
- 5- Mac OS
- 6- Outro

Você foi contratado para desenvolver um programa que leia o resultado da enquete e informe ao final o resultado da mesma. O programa deverá ler os valores até ser informado o valor 0, que encerra a entrada dos dados. Não deverão ser aceitos valores além dos válidos para o programa (0 a 6). Os valores referentes a cada uma das opções devem ser armazenados num vetor. Após os dados terem sido completamente informados, o programa deverá calcular a percentual de cada um dos concorrentes e informar o vencedor da enquete. O formato da saída foi dado pela empresa, e é o seguinte:

Sistema Operacional	Votos	%
-----	-----	---
Windows Server	1500	17%
Unix	3500	40%
Linux	3000	34%
Netware	500	5%
Mac OS	150	2%
Outro	150	2%
-----	-----	---
Total	8800	

O Sistema Operacional mais votado foi o Unix, com 3500 votos, correspondendo a 40% dos votos.



<130>. As Organizações Tabajara resolveram dar um abono aos seus colaboradores em reconhecimento ao bom resultado alcançado durante o ano que passou. Para isto contratou você para desenvolver a aplicação que servirá como uma projeção de quanto será gasto com o pagamento deste abono.

Após reuniões envolvendo a diretoria executiva, a diretoria financeira e os representantes do sindicato laboral, chegou-se a seguinte forma de cálculo:

Cada funcionário receberá o equivalente a 20% do seu salário bruto de dezembro;

O piso do abono será de 100 reais, isto é, aqueles funcionários cujo salário for muito baixo, recebem este valor mínimo;

Neste momento, não se deve ter nenhuma preocupação com colaboradores com tempo menor de casa, descontos, impostos ou outras particularidades.

Seu programa deverá permitir a digitação do salário de um número indefinido (desconhecido) de salários. Um valor de salário igual a 0 (zero) encerra a digitação. Após a entrada de todos os dados o programa deverá calcular o valor do abono concedido a cada colaborador, de acordo com a regra definida acima. Ao final, o programa deverá apresentar:

- O salário de cada funcionário, juntamente com o valor do abono;
  - O número total de funcionário processados;
  - O valor total a ser gasto com o pagamento do abono;
  - O número de funcionário que receberá o valor mínimo de 100 reais;
  - O maior valor pago como abono;
- A tela abaixo é um exemplo de execução do programa, apenas para fins ilustrativos. Os valores podem mudar a cada execução do programa.

Projeção de Gastos com Abono

=====

Salário: 1000

Salário: 300

Salário: 500

Salário: 100

Salário: 4500

Salário: 0



```
Salário      - Abono
R$ 1000.00 - R$  200.00
R$  300.00 - R$  100.00
R$  500.00 - R$  100.00
R$  100.00 - R$  100.00
R$ 4500.00 - R$  900.00
```

```
Foram processados 5 colaboradores
Total gasto com abonos: R$ 1400.00
Valor mínimo pago a 3 colaboradores
Maior valor de abono pago: R$ 900.00
```

<131>. Faça um programa que carregue uma lista com os modelos de cinco carros (exemplo de modelos: FUSCA, GOL, VECTRA etc). Carregue uma outra lista com o consumo desses carros, isto é, quantos quilômetros cada um desses carros faz com um litro de combustível. Calcule e mostre:

- O modelo do carro mais econômico;
- Quantos litros de combustível cada um dos carros cadastrados consome para percorrer uma distância de 1000 quilômetros e quanto isto custará, considerando um que a gasolina custe R\$ 2,25 o litro.

Abaixo segue uma tela de exemplo. A disposição das informações deve ser a mais próxima possível ao exemplo. Os dados são fictícios e podem mudar a cada execução do programa.

#### Comparativo de Consumo de Combustível

```
Veículo 1
Nome: fusca
Km por litro: 7
Veículo 2
Nome: gol
Km por litro: 10
Veículo 3
Nome: uno
Km por litro: 12.5
Veículo 4
Nome: Vectra
Km por litro: 9
Veículo 5
Nome: Peugeot
Km por litro: 14.5
```



#### Relatório Final

```
1 - fusca           -    7.0 - 142.9 litros - R$ 321.43
2 - gol             -   10.0 - 100.0 litros - R$ 225.00
3 - uno             -   12.5 - 80.0 litros - R$ 180.00
4 - vectra          -    9.0 - 111.1 litros - R$ 250.00
5 - peugeot         -   14.5 - 69.0 litros - R$ 155.17
0 menor consumo é do peugeot.
```

<132>. Sua organização acaba de contratar um estagiário para trabalhar no Suporte de Informática, com a intenção de fazer um levantamento nas sucatas encontradas nesta área. A primeira tarefa dele é testar todos os cerca de 200 mouses que se encontram lá, testando e anotando o estado de cada um deles, para verificar o que se pode aproveitar deles. Foi requisitado que você desenvolva um programa para registrar este levantamento. O programa deverá receber um número indeterminado de entradas, cada uma contendo: um número de identificação do mouse o tipo de defeito:

- i. necessita da esfera;
- ii. necessita de limpeza;
- iii. a. necessita troca do cabo ou conector;
- iv. a. quebrado ou inutilizado

Uma identificação igual a zero encerra o programa. Ao final o programa deverá emitir o seguinte relatório:

Quantidade de mouses: 100

Situação	Quantidade	Percentual
1- necessita da esfera	40	40%
2- necessita de limpeza	30	30%
3- necessita troca do cabo ou conector	15	15%
4- quebrado ou inutilizado	15	15%

<119>. Faça um programa que simule um lançamento de dados. Lance o dado 100 vezes e armazene os resultados em um vetor. Depois, mostre quantas vezes cada valor foi conseguido. Dica: use um vetor de contadores (1-6) e uma função para gerar números aleatórios, simulando os lançamentos dos dados.



## Exercícios com funções

<133>. Faça um Programa para uma loja de tintas. O programa deverá pedir o tamanho em metros quadrados da área a ser pintada. Considere que a cobertura da tinta é de 1 litro para cada 6 metros quadrados e que a tinta é vendida em latas de 18 litros, que custam R\$ 80,00 ou em galões de 3,6 litros, que custam R\$ 25,00. Informe ao usuário as quantidades de tinta a serem compradas e os respectivos preços em 3 situações:

- Comprar apenas latas de 18 litros;
- Comprar apenas galões de 3,6 litros;
- Misturar latas e galões, de forma que o desperdício de tinta seja menor.

Acrescente 10% de folga e sempre arredonde os valores para cima, isto é, considere latas cheias.

<134>. Faça um programa para imprimir:

```
1
2  2
3  3  3
....
n  n  n  n  n  n ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n-ésima linha.

<135>. Faça um programa para imprimir:

```
1
1  2
1  2  3
....
1  2  3  ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro imprima até a n-ésima linha.



- <136>. Faça um programa, com uma função que necessite de três argumentos, e que forneça a soma desses três argumentos.
- <137>. Faça um programa, com uma função que necessite de um argumento. A função retorna o valor de caractere 'P', se seu argumento for positivo, e 'N', se seu argumento for zero ou negativo.
- <138>. Faça um programa com uma função chamada somaImposto. A função possui dois parâmetros formais: taxaImposto, que é a quantia de imposto sobre vendas expressa em porcentagem e custo, que é o custo de um item antes do imposto. A função "altera" o valor de custo para incluir o imposto sobre vendas.
- <139>. Faça 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 duas funções: uma para fazer a conversão e uma para a saída. Registre a informação A.M./P.M. como um valor 'A' para A.M. e 'P' para P.M. Assim, a função para efetuar as conversões terá um parâmetro formal para registrar se é A.M. ou P.M. Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar.
- <140>. Faça um programa que use a função valorPagamento para determinar o valor a ser pago por uma prestação de uma conta. O programa deverá solicitar ao usuário o valor da prestação e o número de dias em atraso e passar estes valores para a função valorPagamento, que calculará o valor a ser pago e devolverá este valor ao programa que a chamou. O programa deverá então exibir o valor a ser pago na tela. Após a execução o programa deverá voltar a pedir outro valor de prestação e assim continuar até que seja informado um valor igual a zero para a prestação. Neste momento o programa deverá ser encerrado, exibindo o relatório do dia, que conterá a quantidade e o valor total de prestações pagas no dia. O cálculo do valor a ser pago é feito da seguinte forma. Para pagamentos sem atraso, cobrar o valor da prestação. Quando houver atraso, cobrar 3% de multa, mais 0,1% de juros por dia de atraso.
- <141>. Faça uma função que informe a quantidade de dígitos de um determinado número inteiro informado.
- <142>. Reverso do número. Faça uma função que retorne o reverso de um número inteiro informado. Por exemplo: 127 -> 721.



- <143>. **Jogo de Craps.** Faça um programa de implemente um jogo de Craps. O jogador lança um par de dados, obtendo um valor entre 2 e 12. Se, na primeira jogada, você tirar 7 ou 11, você um "natural" e ganhou. Se você tirar 2, 3 ou 12 na primeira jogada, isto é chamado de "craps" e você perdeu. Se, na primeira jogada, você fez um 4, 5, 6, 8, 9 ou 10,este é seu "Ponto". Seu objetivo agora é continuar jogando os dados até tirar este número novamente. Você perde, no entanto, se tirar um 7 antes de tirar este Ponto novamente.
- <144>. **Data com mês por extenso.** Construa uma função que receba uma data no formato DD/MM/AAAA e devolva uma string no formato D de mesPorExtenso de AAAA. Opcionalmente, valide a data e retorne NULL caso a data seja inválida.
- <145>. **Embaralha palavra.** Construa uma função que receba uma string como parâmetro e devolva outra string com os carateres embaralhados. Por exemplo: se função receber a palavra python, pode retornar npthyo, ophtyn ou qualquer outra combinação possível, de forma aleatória. Padronize em sua função que todos os caracteres serão devolvidos em caixa alta ou caixa baixa, independentemente de como foram digitados.
- <146>. **Desenha moldura.** Construa uma função que desenhe um retângulo usando os caracteres '+' , '-' e '|' '. Esta função deve receber dois parâmetros, linhas e colunas, sendo que o valor por omissão é o valor mínimo igual a 1 e o valor máximo é 20. Se valores fora da faixa forem informados, eles devem ser modificados para valores dentro da faixa de forma elegante.
- <147>. **Quadrado mágico.** Um quadrado mágico é aquele dividido em linhas e colunas, com um número em cada posição e no qual a soma das linhas, colunas e diagonais é a mesma. Por exemplo, veja um quadrado mágico de lado 3, com números de 1 a 9:

8	3	4
1	5	9
6	7	2

Elabore uma função que identifica e mostra na tela todos os quadrados mágicos com as características acima. Dica: produza todas as combinações possíveis e verifique a soma quando completar cada quadrado. Usar um vetor de 1 a 9 parece ser mais simples que usar uma matriz 3x3.





## Exercícios com classes

<148>. Classe Bola: Crie uma classe que modele uma bola:

- a. Atributos: Cor, circunferência, material
- b. Métodos: trocaCor e mostraCor

<149>. Classe Quadrado: Crie uma classe que modele um quadrado:

- a. Atributos: Tamanho do lado
- b. Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área;

<150>. Classe Retangulo: Crie uma classe que modele um retangulo:

- a. Atributos: LadoA, LadoB (ou Comprimento e Largura, ou Base e Altura, a escolher)
- b. Métodos: Mudar valor dos lados, Retornar valor dos lados, calcular Área e calcular Perímetro;

Crie um programa que utilize esta classe. Ele deve pedir ao usuário que informe as medidas de um local. Depois, deve criar um objeto com as medidas e calcular a quantidade de pisos e de rodapés necessárias para o local.

<151>. Classe Pessoa: Crie uma classe que modele uma pessoa:

- a. Atributos: nome, idade, peso e altura
- b. Métodos: Envelhercer, engordar, emagrecer, crescer. Obs:

Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.

<152>. Classe Conta Corrente: Crie uma classe para implementar uma conta corrente. A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo. Os métodos são os seguintes: alterarNome, depósito e saque; No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.

<153>. Classe TV: Faça um programa que simule um televisor criando-o como um objeto. O usuário deve ser capaz de informar o número do canal e aumentar ou diminuir o volume. Certifique-se de que o número do canal e o nível do volume permanecem dentro de faixas válidas.



<154>. Classe Bichinho Virtual: Crie uma classe que modele um Tamagushi (Bichinho Eletrônico):

- a. Atributos: Nome, Fome, Saúde e Idade
- b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade

Obs: Existe mais uma informação que devemos levar em consideração, o Humor do nosso tamagushi, este humor é uma combinação entre os atributos Fome e Saúde, ou seja, um campo calculado, então não devemos criar um atributo para armazenar esta informação por que ela pode ser calculada a qualquer momento.

<155>. Classe Macaco: Desenvolva uma classe Macaco, que possua os atributos nome e bucho (estomago) e pelo menos os métodos comer(), verBucho() e digerir(). Faça um programa ou teste interativamente, criando pelo menos dois macacos, alimentando-os com pelo menos 3 alimentos diferentes e verificando o conteúdo do estomago a cada refeição. Experimente fazer com que um macaco coma o outro. É possível criar um macaco canibal?

<156>. Classe Ponto e Retângulo: Faça um programa completo utilizando funções e classes que:

- a. Possua uma classe chamada Ponto, com os atributos x e y.
- b. Possua uma classe chamada Retangulo, com os atributos largura e altura.
- c. Possua uma função para imprimir os valores da classe Ponto
- d. Possua uma função para encontrar o centro de um Retângulo.
- e. Você deve criar alguns objetos da classe Retangulo.
- f. Cada objeto deve ter um vértice de partida, por exemplo, o vértice inferior esquerdo do retângulo, que deve ser um objeto da classe Ponto.
- g. A função para encontrar o centro do retângulo deve retornar o valor para um objeto do tipo ponto que indique os valores de x e y para o centro do objeto.
- h. O valor do centro do objeto deve ser mostrado na tela
- i. Crie um menu para alterar os valores do retângulo e imprimir o centro deste retângulo.



<157>. Classe Bomba de Combustível: Faça um programa completo utilizando classes e métodos que:

- a. Possua uma classe chamada `bombaCombustível`, com no mínimo esses atributos:
  - i. `tipoCombustivel`.
  - ii. `valorLitro`
  - iii. `quantidadeCombustivel`
- b. Possua no mínimo esses métodos:
  - i. `abastecerPorValor( )` - método onde é informado o valor a ser abastecido e mostra a quantidade de litros que foi colocada no veículo
  - ii. `abastecerPorLitro( )` - método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.
  - iii. `alterarValor( )` - altera o valor do litro do combustível.
  - iv. `alterarCombustivel( )` - altera o tipo do combustível.
  - v. `alterarQuantidadeCombustivel( )` - altera a quantidade de combustível restante na bomba.

OBS: Sempre que acontecer um abastecimento é necessário atualizar a quantidade de combustível total na bomba.

<158>. Classe carro: Implemente uma classe chamada `Carro` com as seguintes propriedades:

- a. Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque.
- b. O consumo é especificado no construtor e o nível de combustível inicial é 0.
- c. Forneça um método `andar( )` que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.
- d. Forneça um método `obterGasolina( )`, que retorna o nível atual de combustível.
- e. Forneça um método `adicionarGasolina( )`, para abastecer o tanque.

Exemplo de uso:

```
meuFusca = Carro(15);           # 15 quilômetros por litro de combustível.
meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.
meuFusca.andar(100);            # anda 100 quilômetros.
meuFusca.obterGasolina()        # Imprime o combustível que resta no tanque.
```



<159>. Classe Conta de Investimento: Faça uma classe `contaInvestimento` que seja semelhante a classe `contaBancaria`, com a diferença de que se adicione um atributo `taxaJuros`. Forneça um construtor que configure tanto o saldo inicial como a taxa de juros. Forneça um método `adicioneJuros` (sem parâmetro explícito) que adicione juros à conta. Escreva um programa que construa uma poupança com um saldo inicial de R\$1000,00 e uma taxa de juros de 10%. Depois aplique o método `adicioneJuros()` cinco vezes e imprime o saldo resultante.

<160>. Classe Funcionário: Implemente a classe `Funcionário`. Um empregado tem um nome (um `string`) e um salário (um `double`). Escreva um construtor com dois parâmetros (nome e salário) e métodos para devolver nome e salário. Escreva um pequeno programa que teste sua classe.

<161>. Aprimore a classe do exercício anterior para adicionar o método `aumentarSalario` (`porcentualDeAumento`) que aumente o salário do funcionário em uma certa porcentagem.

Exemplo de uso:

```
harry=funcionário("Harry",25000)
harry.aumentarSalario(10)
```

<162>. Aprimore Classe `Bichinho Virtual++`: Melhore o programa do bichinho virtual, permitindo que o usuário especifique quanto de comida ele fornece ao bichinho e por quanto tempo ele brinca com o bichinho. Faça com que estes valores afetem quão rapidamente os níveis de fome e tédio caem.

<163>. Crie uma "porta escondida" no programa do bichinho virtual que mostre os valores exatos dos atributos do objeto. Consiga isto mostrando o objeto quando uma opção secreta, não listada no menu, for informada na escolha do usuário. Dica: acrescente um método especial `str()` à classe `Bichinho`.

<164>. Crie uma Fazenda de Bichinhos instanciando vários objetos bichinho e mantendo o controle deles através de uma lista. Imite o funcionamento do programa básico, mas ao invés de exigir que o usuário tome conta de um único bichinho, exija que ele tome conta da fazenda inteira. Cada opção do menu deveria permitir que o usuário executasse uma ação para todos os bichinhos (alimentar todos os bichinhos, brincar com todos os bichinhos,



ou ouvir a todos os bichinhos). Para tornar o programa mais interessante, dê para cada bichinho um nível inicial aleatório de fome e tédio.