

## Questão 1 – Hello World

Crie uma classe em Java chamada `HelloWorld` que imprima a mensagem `"Hello World !!"`.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "HelloWorld.java"

Solução correta!

## Questão 2 – Hello Groot

0.10 / 0.10 

Crie uma classe em Java chamada `HelloGroot` que imprima a mensagem "I am Groot".

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "HelloGroot.java"

Solução correta!

### Questão 3 – Coelho

0.10 / 0.10 

Crie uma classe em Java chamada **Coeelho** que imprima no console a seguinte figura:

$$\begin{pmatrix} \backslash & \backslash \\ (= ' & ' ) \\ ((^ & ) (^ & ) \end{pmatrix}$$

**Dicas:**


- Assim como em outras linguagens, no Java o `"\"` é usado para imprimir caracteres especiais (e.g., `"\n"`, `"\t"`). Para imprimir uma barra, use `"\\"`.
- O rosto tem três linhas. Logo, será necessário executar o método `println` três vezes.
- O Java já imprime uma linha nova ao final do método `println`.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Coelho.java"

Solução correta!

### Questão 4 – Arte ASCII

0.00 / 0.10 

Escreva um programa (classe `ArteAscii`) que imprima uma figura semelhante à representada abaixo, a partir da leitura do número de asteriscos presentes na base. Por exemplo, para uma entrada igual a 5, o resultado produzido será igual ao da figura abaixo.

```
*****
*****
***
**
*
*
**
***
****
*****
```

**Dicas:**

- Para imprimir algo em Java sem a nova linha no final, use o método `System.out.print()`. Exemplo: `System.out.print('*');`
- Para imprimir apenas uma nova linha use `System.out.println();`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "ArteAscii.java"

**Questão 5 – ASCII Art**

0.00 / 0.10 

Escreva um programa (classe `AsciiArt`) que imprima uma figura semelhante à representada abaixo, a partir da leitura do número de asteriscos presentes na base. Por exemplo, para uma entrada igual a 5, o resultado produzido será igual ao da figura abaixo.

```
*****
****  ****
***   ***
**    **
*     *
```

**Dicas:**

- Para imprimir algo em Java sem a nova linha no final, use o método `System.out.print()`. Exemplo: `System.out.print('*');`
- Para imprimir apenas uma nova linha use `System.out.println();`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "AsciiArt.java"

**Questão 6 – Média**

0.00 / 0.10 

Crie uma classe em Java chamada `Media` que calcule e imprima, com duas casas decimais, a

média aritmética de três números reais lidos do teclado.

#### Exemplo de entrada e saída esperada:

- Entrada: `88.0 1.21 70.2`  
Saída: `53.14`

#### Dicas:

- A classe `Scanner`, usada para ler dados do teclado, usa as configurações do sistema para setar o formato dos números reais. Dependendo da linguagem do seu sistema, um número real pode usar "." (ponto) ou "," (vírgula).
- Você pode representar os números reais usando `float` ou `double`.
  - Em Java, um número real isolado (e.g., `1.21`) é considerado como sendo do tipo `double`. Para usar um número como sendo do tipo `float`, use o 'f' ao final do número (e.g., `1.21f`).
  - Java converte automaticamente tipos quando não há perda de precisão. Exemplo: `double nota = 8.7f; // Converte de float para double automaticamente`
  - Entretanto, quando há perda de precisão, isso não é possível e o uso do `cast` é obrigatório. Exemplo: `float nota = (float) 8.7; // Converte de double para float usando o cast`
- Para imprimir algo formatado em Java, use o método `System.out.printf(String format, Object... args)`. Exemplo: `System.out.printf("%.3f\n", media);`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "Media.java"

## Questão 7 – Pintura do Muro

0.00 / 0.10 

Mário precisa pintar um muro, que tem 12m de comprimento e 3m de altura. O material de pintura (galão de tinta, lixa, rolo, etc.) custa R\$100. Cada pintor cobra um preço diferente por m<sup>2</sup> pelo serviço de pintura. Escreva uma classe chamada `PinturaMuro` em Java que, dado o valor cobrado por um pintor (R\$/m<sup>2</sup>), informe o custo total da pintura (com uma casa decimal).

#### Exemplo de entrada e saída esperada:

- Entrada: `3.2`  
Saída: `215.2`

#### Dicas:

- Você pode representar os números reais usando `float` ou `double`.
  - Em Java, um número real isolado (e.g., `1.21`) é considerado como sendo do tipo `double`. Para usar um número como sendo do tipo `float`, use o 'f' ao final do número (e.g., `1.21f`).
  - Java converte automaticamente tipos quando não há perda de precisão. Exemplo: `double nota = 8.7f; // Converte de float para double`

automaticamente

- Entretanto, quando há perda de precisão, isso não é possível e o uso do `cast` é obrigatório. Exemplo: `float nota = (float) 8.7; // Converte de double para float usando o cast`
- A classe `Scanner`, usada para ler dados do teclado, usa as configurações do sistema para setar o formato dos números reais. Dependendo da linguagem do seu sistema, um número real pode usar "." (ponto) ou "," (vírgula).
- Para imprimir algo formatado em Java, use o método `System.out.printf(String format, Object... args)`. Exemplo: `System.out.printf("%.3f\n", media);`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "PinturaMuro.java"

## Questão 8 – Par/Ímpar

0.00 / 0.10 

Crie uma classe em Java chamada `ParImpar` que fique lendo números inteiros do teclado até que o número `-1` seja digitado. Para cada número, imprimir se ele é "PAR" ou "IMPAR".

**Exemplo de entrada e saída esperada:**

- Entrada: 6 67 32 7 -1
- Saída: PAR  
IMPAR  
PAR  
IMPAR

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "ParImpar.java"

## Questão 9 – Temperatura Fahrenheit

0.00 / 0.10 

Escreva uma classe chamada `Fahrenheit` que converta uma temperatura digitada de  $^{\circ}\text{C}$  em  $^{\circ}\text{F}$ . Use apenas uma casa decimal na saída. A fórmula para essa conversão é:

$$F = \frac{9 \times C}{5} + 32$$

**Exemplo de entrada e saída esperada:**

- Entrada: 42
- Saída: 107.6

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Fahrenheit.java"

## Questão 10 – Caixa Eletrônico

0.00 / 0.10 

Um cliente de um banco deseja sacar uma quantia em um caixa eletrônico que possui apenas notas de R\$50, R\$10 e R\$2 disponíveis. Escreva uma classe chamada **CaixaEletronico** que exiba quantas notas de cada tipo devem ser entregues ao cliente. Considere que o cliente pode inserir – intencionalmente ou não – um valor inválido, tal como um número negativo ou um número ímpar. Seu programa deve exibir a mensagem **"Valor Invalido"** nessas situações. A saída do seu programa deve ser de acordo com o exemplo abaixo.

**Exemplo de entrada e saída esperada:**

- Entrada: `378`  
Saída: `7 notas de R$50, 2 notas de R$10 e 4 notas de R$2`

**Dicas:**

- Comece pelas notas de valor mais alto.
- Use o operador de resto da divisão (%) para determinar a quantidade de notas de valor imediatamente mais baixo.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "CaixaEletronico.java"

## Questão 11 – Folha de Pagamento

0.00 / 0.20 

Fernando Silva é programador e ganha por horas trabalhadas. Sabendo que são descontados 11% para o Imposto de Renda e 8% para o INSS. Escreva uma classe chamada **FolhaPagamento** que receba o valor da hora (**double**) e quantidade de horas (**int**) trabalhadas de Fernando e calcule:

- Qual o salário bruto?
- Quanto Fernando pagou de Imposto de Renda?
- Quanto Fernando pagou ao INSS?
- Qual o total de descontos do salário de Fernando?
- Qual é o salário líquido de Fernando?

A saída do programa deve ser de acordo com o exemplo abaixo.

**Exemplo de entrada e saída esperada:**

- Entrada: `50 40`

Saída: Salario bruto: R\$2000.00  
IR: R\$220.00  
INSS: R\$160.00  
Total de descontos: R\$380.00  
Salario liquido: R\$1620.00


**Dica:**

- Salário Bruto – Descontos = Salário Líquido.

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "FolhaPagamento.java"

## Questão 12 – Área do Triângulo

0.00 / 0.20 

Considere um triângulo cujos lados sejam designados por  $a$ ,  $b$  e  $c$ . Considere ainda que  $s = \frac{a+b+c}{2}$ . A área do triângulo pode ser calculada usando a seguinte fórmula:

$$A = \sqrt{s \times (s - a) \times (s - b) \times (s - c)}$$

Escreva um programa em Java (nome da classe: `AreaTriangulo`) que leia os comprimentos dos lados de um triângulo e mostre a sua área com duas casas decimais. Assuma que as entradas podem também corresponder a um triângulo inválido. Neste caso, imprima a mensagem `"Triangulo invalido"`.

**Exemplo de entrada e saída esperada:**

- Entrada: 4 8 6  
Saída: 11.62

**Dicas:**

- Em um triângulo, a soma dos comprimentos de quaisquer dois lados é maior que o comprimento do terceiro lado [Wiki].
- Para calcular uma raiz quadrada em Java, usa-se o método `sqrt(double a)` da classe `Math`. Exemplo: `double raiz = Math.sqrt(24601);`
- Para imprimir algo formatado em Java, use o método `System.out.printf(String format, Object... args)`. Exemplo: `System.out.printf("%.3f\n", media);`
- Você pode representar os números reais usando `float` ou `double`.
  - Em Java, um número real isolado (e.g., 1.21) é considerado como sendo do tipo `double`. Para usar um número como sendo do tipo `float`, use o `'f'` ao final do número (e.g., 1.21f).
  - Java converte automaticamente tipos quando não há perda de precisão. Exemplo: `double nota = 8.7f; // Converte de float para double automaticamente`
  - Entretanto, quando há perda de precisão, isso não é possível e o uso do `cast` é

obrigatório. Exemplo: `float nota = (float) 8.7; // Converte de double para float usando o cast`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AreaTriangulo.java"

## Questão 13 – Números Narcisistas

0.00 / 0.20

Um **número narcisista** é um número que segue a seguinte regra:

$$n = d_k^k + d_{k-1}^k + \dots + d_2^k + d_1^k,$$

onde  $k$  é a quantidade de dígitos do número. Por exemplo, o número **153** é um número narcisista, pois  $153 = 1^3 + 5^3 + 3^3$ .

Desenvolva um programa (classe **NumeroNarcisista**) que leia um número inteiro do teclado e diga **"SIM"** se ele for um número narcisista ou **"NAO"**, caso contrário.

**Dicas:**

- Use o operador de resto da divisão (%).
- Para calcular uma exponenciação em Java, usa-se o método `pow(double a, double b)` da classe **Math**. Exemplo: `double exp = Math.sqrt(2, 8);`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "NumeroNarcisista.java"

## Questão 14 – Área do Círculo, Volume da Esfera

0.00 / 0.20

Escreva um programa (classe **AreaVolume**) que leia o valor de um raio  $r$ , inserido a partir do teclado. O programa deverá mostrar a área de um círculo com o raio  $r$  e o volume de uma esfera com raio  $r$ , de acordo com o exemplo abaixo. **Para esta questão, use apenas variáveis do tipo `double`.**

**Exemplo de entrada e saída esperada:**

- Entrada: `45`

Saída:

```
Um circulo com raio de 45.00 centimetros tem uma area de 6361.73
centimetros quadrados.
```

```
Uma esfera com raio de 45.00 centimetros tem um volume de 381703.51
centimetros cubicos.
```

**Fórmulas:**

- A área de um círculo é dada pela fórmula  $A = \pi r^2$ .
- O volume de uma esfera é dada pela fórmula  $V = \frac{4}{3}\pi r^3$ .

**Dicas:**

- Use apenas variáveis do tipo `double` para não perder precisão e não dar erro na correção automática.
- Use a constante `PI` presente na classe `Math` do Java para pegar o valor de  $\pi$ . Exemplo:  
`double pi = Math.PI;`
- Note que em Java,  $1/3 = 1$ , enquanto que  $1.0/3.0 = 1.333$ .
- Para calcular uma exponenciação em Java, usa-se o método `pow(double a, double b)` da classe `Math`. Exemplo: `double exp = Math.sqrt(2, 8);`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AreaVolume.java"

## Questão 15 – Ponto e Reta

0.00 / 0.20 

Faça um programa (classe `PontoReta`) para indicar se um ponto  $P(x, y)$  do plano cartesiano pertence à reta  $2x + y = 3$ . Sua saída deverá ser de acordo com os exemplos abaixo.


**Exemplos de entrada e saída esperada:**

- Entrada: `0.5 2.0`  
Saída: `Ponto (0.5, 2.0) pertence a reta 2x + y = 3.`
- Entrada: `6.0 3.8`  
Saída: `Ponto (6.0, 3.8) nao pertence a reta 2x + y = 3.`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "PontoReta.java"

## Questão 16 – Desconto

0.00 / 0.20 

Para atrair mais clientes, uma loja de roupas oferece um desconto de 5% em compras de R\$200,00 ou mais. Escreva um programa (classe `Desconto`) que lê o preço sem desconto de uma compra e imprime o valor a ser pago pelo cliente (com duas casas decimais).

**Exemplo de entrada e saída esperada:**

- Entrada: `320.99`  
Saída: `304.94`



**Dica:**

- A classe `Scanner`, usada para ler dados do teclado, usa as configurações do sistema para setar o formato dos números reais. Dependendo da linguagem do seu sistema, um número real pode usar "." (ponto) ou "," (vírgula.....).
- Você pode representar os números reais usando `float` ou `double`.
  - Em Java, um número real isolado (e.g., `1.21`) é considerado como sendo do tipo `double`. Para usar um número como sendo do tipo `float`, use o 'f' ao final do número (e.g., `1.21f`).
  - Java converte automaticamente tipos quando não há perda de precisão. Exemplo:  
`double nota = 8.7f; // Converte de float para double automaticamente`
  - Entretanto, quando há perda de precisão, isso não é possível e o uso do `cast` é obrigatório. Exemplo: `float nota = (float) 8.7; // Converte de double para float usando o cast`
- Para imprimir algo formatado em Java, use o método `System.out.printf(String format, Object... args)`. Exemplo: `System.out.printf("%.3f\n", media);`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Desconto.java"

**Questão 17 – Angry Birds**

0.00 / 0.20

Considere o cenário do jogo *Angry Birds* (figura abaixo), onde um pássaro é lançado com uma velocidade inicial  $v_0$  a partir do estilingue, cujo elástico faz um ângulo  $\alpha$  com o solo.



Figura: exemplo de trajetória. Fonte: [polibentinhofisica.blogspot.com](http://polibentinhofisica.blogspot.com)

Considere ainda que o pássaro e o porco-alvo estão na mesma altura em relação ao solo. O alcance máximo ( $R$ ) do pássaro horizontalmente é dado pela seguinte equação:

$$R = \frac{|v_o|^2 \cdot \sin(2\alpha)}{g}$$

onde  $|v_o|$  é o valor em módulo da velocidade inicial (sem considerar os componentes vertical ou horizontal) e  $g = 9.8m/s^2$  é a aceleração da gravidade.

Escreva um programa (classe `AngryBirds`) que leia a velocidade inicial  $v_o$ , o ângulo  $\alpha$  (em graus), e a distância horizontal  $D$  entre o pássaro e o porco, e informe se o pássaro atingirá (saída 1) ou não o porco (saída 0). Admita uma tolerância de 0.1. Ou seja, se  $D = 20$  e  $R = 19.9$ , então podemos considerar que o pássaro acerta o porco.

#### Exemplos de entrada e saída esperada:

- Entrada: 30 45 91.83  
Saída: 1
- Entrada: 30 45 100  
Saída: 0

#### Dicas:

- Funções trigonométricas do Java operam em radianos. Como resultado, você vai precisar converter a entrada do usuário de graus para radianos antes de calcular a distância. Para isso, use o método `toRadians(double angdeg)` da classe `Math`. Exemplo:  
`double radians = Math.toRadians(45);`
- Use os métodos trigonométricos da classe `Math` para calcular o seno – `Math.sin(double a)`, cosseno – `Math.cos(double a)` e arco cosseno – `Math.acos(double a)`.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AngryBirds.java"

## Questão 18 – Tanque de Combustível

0.00 / 0.20 

Um tanque de combustível tem o formato esférico. Escreva um programa (classe `TanqueCombustivel`) que leia o valor do raio ( $r$ ) do tanque, o valor da altura do ar ( $x$ ) e a opção para saber se deseja calcular o volume de ar (opção 1) ou o volume de combustível no tanque (opção 2).

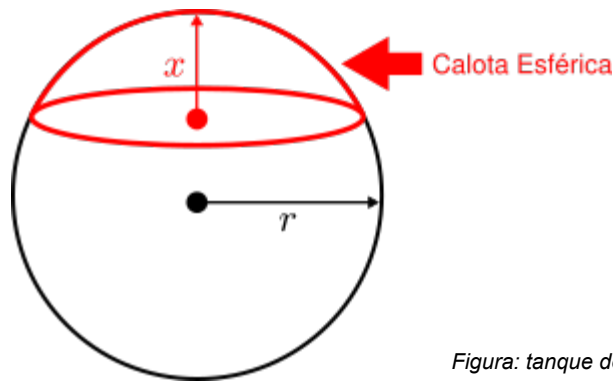


Figura: tanque de combustível

A saída do programa deverá ser o volume conforme a opção selecionada, com *quatro* casas decimais de precisão. Não é necessário verificar situações inválidas (opções diferentes de 1 ou 2, ou valores de altura maiores que o raio da esfera).

#### Fórmulas:

- Volume da esfera de raio  $r$ :  $V = \frac{4}{3}\pi r^3$
- Volume da calota esférica de raio  $r$  e altura  $x$ :  $V = \frac{\pi}{3}x^2(3r - x)$

#### Exemplos de entrada e saída esperada:

- Entrada: 30 27 1  
Saída: 48094.6419
- Entrada: 30 27 2  
Saída: 65002.6936

#### Dica:

- O volume de ar corresponde à fórmula da calota esférica. Já o volume do combustível corresponde ao complemento do volume da calota esférica em relação ao volume total da esfera.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "TanqueCombustivel.java"

## Questão 19 – Animais nas Cédulas do Real

0.00 / 0.20

As cédulas do real começaram a entrar em circulação no Brasil em 1994. Diferentemente das moedas que haviam circulado anteriormente, o real não traz na sua nota personalidades da história nacional, mas sim animais da fauna brasileira, conforme mostra a tabela abaixo.

| Animal    | Valor da Cédula |
|-----------|-----------------|
| Tartaruga | R\$ 2           |
| Garça     | R\$ 5           |

|                   |         |
|-------------------|---------|
| Arara             | R\$ 10  |
| Mico-leão-dourado | R\$ 20  |
| Onça-pintada      | R\$ 50  |
| Garoupa           | R\$ 100 |

Escreva um programa (classe **AnimaisCedulas**) que leia o valor de uma cédula e apresente na tela o nome do animal representado no verso da nota. Se não existir uma cédula no valor inserido, o programa deverá imprimir **"erro"**.

**Dica:**

- A saída é com acentos e com a primeira letra em maiúsculo.

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "AnimaisCedulas.java"

## Questão 20 – Conta de Energia

0.00 / 0.20 

Escreva um programa (classe **ContaEnergia**) que determine o preço a pagar pelo fornecimento de energia elétrica. Como entrada, ele deve receber o consumo de energia (em kWh) e o tipo de instalação (**R** para residências, **I** para indústrias, e **C** para comércios). Use a tabela a seguir para calcular o preço devido:

| Preço por tipo e faixa de consumo |               |               |
|-----------------------------------|---------------|---------------|
| Tipo                              | Faixa (kWh)   | Preço         |
| Residencial                       | Até 500       | R\$ 0,40      |
|                                   | Acima de 500  | R\$ 0,65      |
| Comercial                         | Até 1000      | R\$ 0,55      |
|                                   | Acima de 1000 | R\$ 0,.....60 |
| Industrial                        | Até 5000      | R\$ 0,55      |
|                                   | Acima de 5000 | R\$ 0,60      |

A saída deve ter duas casas decimais. Caso algum valor inválido seja inserido, o programa deve imprimir **-1.00**.

**Exemplo de entrada e saída esperada:**

- Entrada: **6340 R**  
Saída: **4121.00**

**Dica:**

- A classe **Scanner** não possui o método `nextChar()`. Para ler um caractere do teclado use: `char tipo = scan.next().charAt(0);`

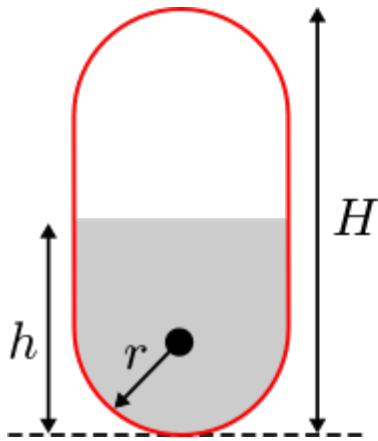
*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "ContaEnergia.java"

## Questão 21 – Volume de Combustível

0.00 / 0.30

Considere um tanque de combustível com o formato descrito na figura abaixo.



$H$  - Altura total do tanque  
 $h$  - Nível de combustível no tanque  
 $r$  - raio dos bojos semiesféricos inferior e superior

Figura: tanque de combustível

Escreva um programa (classe `VolumeCombustivel`) que calcule o volume de combustível (em litros), com três casas decimais de precisão, dadas as medidas (em metros) de altura total do tanque ( $H$ ), nível de combustível no tanque ( $h$ ) e raio dos bojos semi-esféricos ( $r$ ). Caso algum dos dados sejam inválidos, a saída deve ser `-1.000`.

**Fórmulas:**

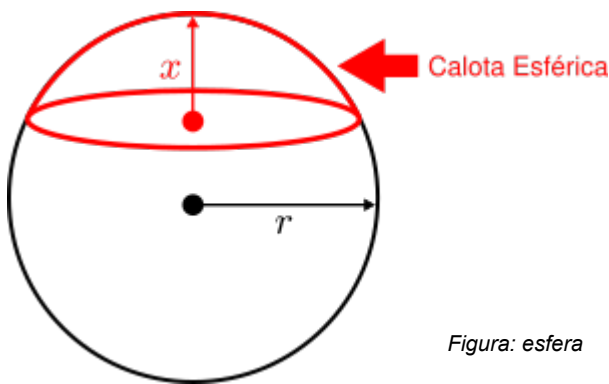


Figura: esfera

- Volume da esfera de raio  $r$ :  $V = \frac{4}{3}\pi r^3$
- Volume da calota esférica de raio  $r$  e altura  $x$ :  $V = \frac{\pi}{3}x^2(3r - x)$
- Volume do cilindro de raio  $r$  e altura  $x$ :  $V = \pi r^2 x$

**Dicas:**

- Verifique as diversas possibilidades do nível de combustível  $h$  em relação à altura do tanque  $H$ .

- Desenhe! Use papel e caneta para melhor visualizar as diversas variáveis e diferentes casos.
- Use a constante `PI` presente na classe `Math` do Java para pegar o valor de  $\pi$ . Exemplo:  
`double pi = Math.PI;`
- Para calcular uma exponenciação em Java, usa-se o método `pow(double a, double b)` da classe `Math`. Exemplo: `double exp = Math.sqrt(2, 8);`


**Exemplo de entrada e saída esperada:**

- Entrada: `50 40 10`  
Saída: `11519.173`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "VolumeCombustivel.java"

## Questão 22 – Soma de Várias Coleções

0.00 / 0.30 

Escreva um programa (classe `SomaColecoes`) que calcula a soma de várias coleções de valores digitados pelo usuário. O usuário irá inserir `-1` para indicar o fim de uma coleção e o início da próxima. Uma coleção sem elementos indica o final do programa.

**Exemplo de entrada e saída esperada:**

- Entrada: `20 30 40 -1 8 7 64 -1 38 26 15 95 -1 -1`  
Saída: `90`  
`79`  
`174`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "SomaColecoes.java"

## Questão 23 – Média de uma Coleção

0.00 / 0.30 

Escreva um programa (classe `MediaColecao`) que calcula a média de uma coleção de valores digitados pelo usuário, com precisão de duas casas decimais. O usuário irá inserir `-1` para indicar que não há mais valores serem fornecidos.....

**Exemplo de entrada e saída esperada:**

- Entrada: `5 3 9 3 1 -1`  
Saída: `4.20`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "MediaColecao.java"

## Questão 24 – Média de Várias Coleções

0.00 / 0.30

Escreva um programa (classe `MediaColecoes`) que calcula a média de várias coleções de valores digitados pelo usuário, com duas casas decimais. O usuário irá inserir `-1` para indicar o fim de uma coleção e o início da próxima. Uma coleção sem elementos indica o final do programa.

**Exemplo de entrada e saída esperada:**

• Entrada: 20 30 40 -1 8 7 64 -1 38 26 15 95 -1 -1  
Saída: 30.00  
26.33  
43.50

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "MediaColecoes.java"

## Questão 25 – Aproximação de $\pi$

0.00 / 0.30

O valor de  $\pi$  pode ser aproximado pela seguinte série infinita:

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \dots$$

Escreva um programa (classe `AproximacaoPi`) .....que exibe  $N$  aproximações de  $\pi$ , sendo  $N$  um número natural inserido pelo usuário. A primeira aproximação deve fazer uso de apenas o primeiro termo da série infinita (número 3). Cada aproximação adicional indicada pelo seu programa deve incluir mais um termo na série, fazendo uma melhor aproximação de  $\pi$  cada vez que um termo é incluído na soma. Utilize até seis casas decimais de precisão. **Para esta questão, use apenas variáveis do tipo `double`.**

**Dica:**

- Use apenas variáveis do tipo `double` para não perder precisão e não dar erro na correção automática.
- Determine o termo geral da série antes de começar a programar.

**Exemplos de entrada e saída esperada:**

• Entrada: 1

```


Saída: 3.000000
• Entrada: 2
Saída: 3.000000
3.166667
• Entrada: 5
Saída: 3.000000
3.166667
3.133333
3.145238
3.139683

```

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AproximacaoPi.java"

## Questão 26 – Aproximação do .....Seno

0.00 / 0.30 

O seno de um ângulo  $x$ , medido em *radianos*,  $\pi \leq x \leq \pi$ , pode ser calculado através da seguinte [série de Taylor](#):

$$\text{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

Escreva um programa (classe `AproximacaoSeno`) que leia um ângulo, medido em *graus*, e o número  $k$  de termos da série, nessa ordem, e imprima o valor da série com dez casas decimais de precisão para cada um dos termos, de 1 a  $k$ . **Para esta questão, use apenas variáveis do tipo `double`.**

### Dicas:

- Use apenas variáveis do tipo `double` para não perder precisão e não dar erro na correção automática.
- Determine o termo geral da série antes de começar a programar.
- Use o `double`, devido à precisão pedida.
- Para converter a entrada do usuário de graus para radianos use o método `toRadians(double angdeg)` da classe `Math`. Exemplo:  
`double radians = Math.toRadians(45);`
- Java não possui uma implementação do cálculo de fatorial. Você precisará implementar dentro do seu `main` ou criar um [método](#) (função) para fazer isso.

### Exemplos de entrada e saída esperada:

```

• Entrada: 45 2
Saída: 0.7853981634
0.7046526512
• Entrada: 90 8

```



Saída: 1.....5707963268  
0.9248322293  
1.0045248555  
0.9998431014  
1.0000035426  
0.9999999437  
1.0000000007  
1.0000000000

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AproximacaoSeno.java"

## Questão 27 – Valor do Expoente

0.00 / 0.30 

Faça um programa (classe `ValorExpoente`) que use um laço `while` para determinar qual o menor valor de  $k$  necessário para que o valor da seguinte soma ultrapasse um número informado pelo usuário via teclado:

$$2^1 + 2^2 + 2^3 + \dots + 2^k$$

Exemplos de entrada e saída esperada:

- Entrada: 1  
Saída: 1
- Entrada: 2  
Saída: 2
- Entrada: 6  
Saída: 3

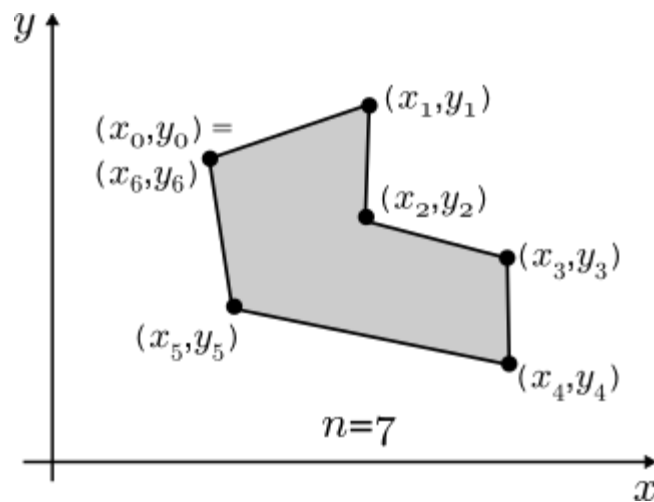
O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "ValorExpoente.java"

## Questão 28 – Área do Polígono

0.00 / 0.30 

Se  $n$  pontos estão ligados formando um polígono fechado, como mostrado abaixo,



a área  $A$  do polígono pode ser determinada por:

$$A = \frac{1}{2} \left| \sum_{i=0}^{n-2} (x_{i+1} + x_i)(y_{i+1} - y_i) \right|$$

Observe que, embora o polígono ilustrado acima tenha apenas seis vértices distintos, temos que  $n = 7$  para este polígono, pois o algoritmo espera que o último ponto,  $(x_6, y_6)$ , seja uma repetição do ponto inicial,  $(x_0, y_0)$ .

Escreva um programa (classe `AreaPoligono`) .....que receba como entrada as coordenadas  $(x, y)$  dos vértices do polígono como dois vetores de valores reais, um contendo os valores das abscissas  $(x)$  e o outro vetor contendo os valores das respectivas ordenadas  $(y)$ , ambos finalizados através do número `-1`. Na saída, deve-se informar a área do polígono, com quatro casas decimais de precisão.

**Para simplificar o código, considere que:**

- Os vetores fornecidos são sempre do mesmo tamanho;
- Cada vetor possui pelo menos quatro elementos (o que garante pelo menos a existência de um triângulo); e
- O primeiro e o último elementos de cada vetor de entrada são sempre iguais.

**Dicas:**

- Para aplicar a função módulo no resultado do somatório, use o método `abs(double a)` da classe `Math`. Exemplo: `double sumAbs = Math.abs(sum);`
- Note que em Java, `1/3 = 1`, enquanto que `1.0/3.0 = 1.333`.
- Atenção para a condição do somatório: menor ou igual a  $n - 2$ .

**Exemplos de entrada e saída esperada:**

- Entrada: `4 4 7 7 9 7 4 -1 0 7.5 7.5 3 0 0 0 -1`  
Saída: `25.5000`
- Entrada: `4 9 11 2 4 -1 10 7 2 2 10 -1`  
Saída: `45.5000`

**Fonte:**

- Hanly & Koffman. "Problem Solving and Program Design in C", 5th edition. Pearson, 2007. (p. 423)

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "AreaPoligono.java" .....

## Questão 29 – Data por Extenso

0.00 / 0.30 

Escreva um programa em Java (classe `DataExtenso`) que leia do teclado uma data no formato "`ddmmaaaa`" e imprima essa data por extenso (suponha valores sempre válidos).

**Exemplo de entrada e saída esperada:**

- Entrada: `21102015`  
Saída: `21 de outubro de 2015`

**Dicas:**


- Existem diversas formas de se isolar os números. Sugestão - implemente e teste as duas formas diferentes abaixo:
  1. Leia a data como uma `String` e use o método `substring(int beginIndex, int endIndex)` da classe `String` para isolar os valores e o método `parseInt(String s)` da classe `Integer` para converter de `String` para `int`. Exemplo:

```
String valorStr = "24601";
String numStr = valorStr.substring(2, 5); // Pega os caracteres
2 a 5 "601"
int num = Integer.parseInt(numStr);      // Converte a String
"601" para o int 601
```
  2. Leia a data como um `int` e use operações de divisão e resto para isolar o dia/mês/ano. Esta é a forma mais eficiente.

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "DataExtenso.java" .....

## Questão 30 – Operações Básicas em uma String

0.00 / 0.30 

Escreva um programa (classe `OperacoesString`) que leia uma string digitada pelo usuário. Em seguida, imprima as seguintes informações sobre a string, necessariamente na ordem indicada:

1. Quantidade de caracteres da string
2. Primeiro caractere da string

3. Último caractere da string
4. Conversão da string em caracteres maiúsculos
5. Conversão da string em caracteres minúsculos
6. Substituição de todas as letras "a" por "e"
7. Impressão dos caracteres com índices pares
8. Quantidade de vogais na string

**Dica:**

- Use a [documentação oficial da classe String](#) para ver os diversos métodos que você pode executar para conseguir as informações acima. Por exemplo, para pegar a quantidade de caracteres, a classe `String` tem o método `length()`, como mostrado a seguir:

```
String nome = "Jebediah Kerman";
int tamNome = nome.length();
```
- Para ler uma string que possui espaços, use o método `nextLine()` da classe `Scanner`. Exemplo: `String linha = scan.nextLine();`
- Note que, em Java, uma `String` usa aspas duplas (e.g., `"string"`), enquanto que um `char` usa aspas simples (e.g., `'c'`).
- Ao contar a quantidade de vogais, não .....esqueça de contar também as vogais em maiúsculas.

**Exemplo de entrada e saída esperada:**

- Entrada: `Jebediah Kerman`  
Saída: `15`  
`J`  
`n`  
`JEBEDIAH KERMAN`  
`jebediah kerman`  
`Jebedieh Kermen`  
`Jbda emn`  
`6`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "OperacoesString.java"

## Questão 31 – Cifra de César

0.00 / 0.40 

A criptografia tem por objetivo esconder o significado de uma mensagem para um possível interceptador. Para tanto, os caracteres do texto podem ser misturados de acordo com um protocolo (algoritmo) previamente estabelecido entre emissor e receptor. Assim, o receptor da mensagem pode reverter o protocolo misturador e tornar a mensagem compreensível.

Uma maneira de se criptografar uma mensagem é substituir o "alfabeto original" por um "alfabeto cifrado", deslocado por um determinado número de letras (5, no exemplo abaixo), em

relação ao alfabeto original. O registro mais antigo desse procedimento é atribuído ao general romano **Júlio César** (100-44 a.C.). Por isso, ele é conhecido como **Cifra de César**. Por convenção, escreve-se o alfabeto original em minúsculas e o alfabeto cifrado em maiúsculas.

#### Exemplo da Cifra de César (Código: 5):

- Alfabeto original: `a b c d e f g h i j k l m n o p q r s t u v w x y z`
- Alfabeto cifrado: `F G H I J K L M N O P Q R S T U V W X Y Z A B C D E`
- Texto original: `v o v o v i u a u v a`
- Texto cifrado: `A T A T A N Z F Z A F`

Escreva um programa (classe **CifraCesar**) que leia o valor do deslocamento (tal como 5 no exemplo acima) e depois leia um texto a ser codificado. A saída deve ser o texto criptografado segundo a Cifra de César explicada acima.

Independentemente dos caracteres de entrada, a saída deverá ser apresentada em caracteres MAIÚSCULOS. Durante a criptografia, ignore qualquer caractere que não seja uma letra do alfabeto, tais como números, pontuação e caracteres com acentos. Estes deverão ser repetidos na saída tal qual foram inseridos na entrada.

#### Exemplo de entrada e saída esperada:

- Entrada: `12 it's an older code, sir, but it checks out`  
Saída: `UF'E MZ AXPQD OAPQ, EUD, NGF UF OTQOWE AGF`
- Entrada: `4 well... here we are all together for the first time`  
Saída: `AIPP... LIVI AI EVI EPP XSKIXLIV JSV XLI JMVWX XMQI`

#### Dicas:

- Note que, em Java, uma **String** usa aspas duplas (e.g., "string"), enquanto que um **char** usa aspas simples (e.g., 'c').
- Para ler uma string que possui espaços, use o método `nextLine()` da classe **Scanner**. Exemplo: `String linha = scan.nextLine();`
- Para pegar um caractere de uma **String** usando seu índice, use o método `charAt(int index)` da classe **String**. Exemplo: `char caract = textoLimpo.charAt(i);`
- Assim como em outras linguagens, um **char** em Java é um número inteiro de acordo com a **Tabela ASCII**. Você pode fazer:
  - operações matemáticas. Exemplo: `caract - 'a';`
  - comparações de um caractere com um número. Exemplo: `caract > 65`
  - comparações entre caracteres. Exemplo: `caract > 'a'`
- Uma maneira de encontrar a letra do alfabeto cifrado que vai além da letra 'Z' é utilizando o resto da divisão com o número total de letras do alfabeto. Exemplo:

```
char cCifrado;
cCifrado = (char) (cOriginal ..... - 'a');           // Conv. de char para
int entre 0-26
cCifrado = (char) ((cCifrado + desloc) % 26); // Pega a letra cifrada
como int entre 0-26
cCifrado += 'A';                                     // Conv. de int entre
0-26 para char entre A-Z
```

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "CifraCesar.java"

## Questão 32 – Palíndromos

0.00 / 0.40 

Um palíndromo é uma sequência 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, espaços, acentos e pontuação são ignorados. A frase "SUBI NO ONIBUS" é o exemplo de uma frase palíndroma onde os espaços foram ignorados.

Escreva um programa (classe `Palindromos`) que leia uma sequência de caracteres. Em seguida, o programa deve imprimir a **mesma** sequência em letras maiúsculas com os espaços removidos e informar se ela é um palíndromo (saída `1`) ou não (saída `0`).

**Exemplos de entrada e saída esperada:**

- Entrada: `A Rara Arara`  
Saída: `ARARAARARA 1`
- Entrada: `Tecnicas Avancadas de Programacao`  
Saída: `TECNICASAVANCADASDEPROGRAMACAO 0`
- Entrada: `a torre da derrota`  
Saída: `ATORREDADERROTA 1`

**Dicas:**

- Para ler uma string que possui espaços, use o método `nextLine()` da classe `Scanner`. Exemplo: `String linha = scan.nextLine();`
- Para converter uma string para maiúscula, use o método `toUpperCase()` da classe `String`. Exemplo: `String linhaMaiusc = linha.toUpperCase();`
- Use o método `replace(CharSequence target, CharSequence replacement)` da classe `String` para remover os espaços. Exemplo: `linha = linha.replace(" ", "");`
- Para pegar um caractere de uma `String` usando seu índice, use o método `charAt(int index)` da classe `String`. Exemplo: `char caract = textoLimpo.charAt(i);`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "Palindromos.java"

## Questão 33 – Operações em Números Inteiros

0.00 / 0.40 

Escreva programa em Java (classe `OperacoesInteiros`) que leia do teclado um vetor de inteiros. Em seguida, o programa deve imprimir na ordem:

- Quantidade de elementos

- Quantos são pares
- Quantos são ímpares
- Soma total
- Média (duas casas decimais)
- Maior
- Menor

O final de um vetor é determinado pelo número `-1`. Seu programa deve repetir esse procedimento indefinidamente para diversos vetores, de tamanhos variados, até que um vetor sem elementos seja inserido, terminando a execução do programa.

**Exemplo de entrada e saída esperada:**

```
• Entrada: 1 5 2 8 4 -1 10 54 23 78 -1 -1
Saída: 5
      3
      2
      20
      4.00
      8
      1
      4
      3
      1
      165
      41.25
      78
      10
```

**Dica:**

- A classe `Integer` possui as constantes `MAX_VALUE` e `MIN_VALUE` para o maior e menor número inteiro armazenável. Exemplo: `int numero = Integer.MAX_VALUE;`

*O prazo de entrega do trabalho terminou. ....Portanto, o botão abaixo está desabilitado.*

Enviar "OperacoesInteiros.java"

## Questão 34 – Time de Futebol

0.00 / 0.40 

Os resultados das partidas de um time de futebol ao longo de um campeonato são armazenados em dois vetores de mesmo tamanho. O primeiro vetor guarda o número de gols efetuados por esse time em cada partida. O segundo vetor guarda o número de gols efetuados pelo time adversário. Uma posição  $i$  de cada vetor indica a  $i$ -ésima partida realizada.

Escreva um programa (classe `TimeFutebol`) que leia esses dois vetores, na ordem em que foram explicados. Como saída, deve ser informado em uma única linha os seguintes dados, em ordem, em relação ao time:

- Número de vitórias
- Número de empates
- Número de derrotas

Considere que o final de um vetor é indicado pelo número **-1** e as entradas fornecidas são sempre válidas, *i.e.*, o tamanho dos dois vetores de gols são iguais e cada elemento do vetor, que indica o número de gols, é um inteiro não-negativo.

**Exemplo de entrada e saída esperada:**

- Entrada: `4 0 2 3 1 3 -1 0 0 2 5 1 2 -1`  
Saída: `2 3 1`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "TimeFutebol.java"

## Questão 35 – Aprovação em Disciplina

0.00 / 0.40 

Na Universidade Federal do Amazonas, um aluno é *aprovado* em uma disciplina se atender a dois critérios: ter média igual ou superior a 5.0 e ter frequência igual ou superior a 75% da carga horária. Se sua frequência for menor que esse limiar, o aluno é *reprovado por frequência*, independentemente da sua nota. Por fim, o aluno é *reprovado por nota* se, tendo comparecido ao número mínimo de aulas, não atingiu a média exigida.

As notas dos alunos de uma classe são guardadas em um vetor de reais. A quantidade de presenças às aulas (em horas) é armazenada em outro vetor de mesmo tamanho, mas contendo elementos inteiros.

Escreva um programa (classe `AprovacaoDisciplina`) que leia o vetor de notas (reais), o vetor de presença (inteiros) e a carga horária da disciplina (inteiro), nessa ordem, usando o número **-1** para indicar o final de um vetor. Como saída, deve ser informado, em uma única linha, os seguintes dados em ordem:

- Número de alunos aprovados
- Número de alunos reprovados por nota
- Número de alunos reprovados por frequência

Considere que as entradas fornecidas são sempre válidas: notas entre 0.0 e 10.0, frequência entre 0 e a carga horária, tamanhos iguais dos vetores de notas e frequência.

**Exemplo de entrada e saída esperada:**

- Entrada: `10.0 10.0 9.0 -1 44 45 46 -1 60`  
Saída: `2 0 1`

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*



Enviar "AprovacaoDisciplina.java"

## Questão 36 – Faltas ao Trabalho

0.00 / 0.40

O setor de Recursos Humanos (RH) de uma empresa está preocupado com as faltas de seus empregados ao trabalho. Como estudo inicial, o RH deseja determinar quantas faltas acontecem em cada dia de semana de trabalho (segunda a sábado).

Um vetor armazena o histórico de faltas registradas na empresa. Cada posição armazena um número que representa o dia da semana em que algum empregado faltou ao trabalho. O número 2 indica que houve uma falta na segunda-feira, 3 indica uma falta em dia de terça-feira, e assim por diante, até o número 7, que indica uma falta em dia de sábado.

Escreva um programa (classe `FaltasTrabalho`) que leia esse vetor, de tamanho qualquer e finalizado pelo número -1. Na saída, o programa deve imprimir seis números contendo o percentual de faltas que aconteceram na segunda, terça, ..., até sábado, nesta ordem, com uma casa decimal de precisão. Considere que a entrada fornecida é sempre válida.

**Exemplo de entrada e saída esperada:**

- Entrada: 2 2 2 3 3 7 7 7 7 7 -1
- Saída: 30.0 20.0 0.0 0.0 0.0 50.0

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*

Enviar "FaltasTrabalho.java"

## Questão 37 – Horas de Trabalho

0.00 / 0.40

Suponha que as horas de trabalho semanais dos funcionários de uma empresa são armazenadas em uma tabela semelhante à do exemplo abaixo. Cada linha registra o número de horas trabalhadas por um funcionário em sete colunas, uma para cada dia da semana, de domingo a sábado. Por exemplo, a tabela a seguir armazena as horas de trabalho para quatro funcionários.

|               | DOM | SEG | TER | QUA | QUI | SEX | SAB |
|---------------|-----|-----|-----|-----|-----|-----|-----|
| Funcionário 0 | 2   | 4   | 3   | 4   | 5   | 8   | 8   |
| Funcionário 1 | 7   | 3   | 4   | 3   | 3   | 4   | 4   |
| Funcionário 2 | 3   | 3   | 4   | 3   | 3   | 2   | 2   |
| Funcionário 3 | 9   | 3   | 4   | 7   | 3   | 4   | 1   |

Escreva um programa (classe `HorasTrabalho`) que leia uma matriz  $N \times 7$ , sendo  $N > 1$ , e imprima quantas horas cada um dos funcionários trabalhou durante a semana, um por linha. Considere o número -1 como o final da entrada, ou seja, os 7 primeiros números correspondem ao **Funcionário 0**, seguido dos 7 números para o **Funcionário 1**, e assim por diante.

**Exemplo de entrada e saída esperada:**

|            |  |
|------------|--|
| • Entrada: | 2 4 3 4 5 8 8 7 3 4 3 3 4 4 3 3 4 3 3 2 2 9 3 4 7 3 4 1 -1 |
| Saída:     | 34   |
|            | 28   |
|            | 20   |
|            | 31   |

*O prazo de entrega do trabalho terminou. ....Portanto, o botão abaixo está desabilitado.*

Enviar "HorasTrabalho.java"

## Questão 38 – Dia da Semana

0.00 / 0.40

Suponha que as horas de trabalho semanais dos funcionários de uma empresa são armazenadas em uma tabela semelhante à do exemplo abaixo. Cada linha registra o número de horas trabalhadas por um funcionário em sete colunas, uma para cada dia da semana, de domingo a sábado. Por exemplo, a tabela a seguir armazena as horas de trabalho para quatro funcionários.

|               | DOM | SEG | TER | QUA | QUI | SEX | SAB |
|---------------|-----|-----|-----|-----|-----|-----|-----|
| Funcionário 0 | 2   | 4   | 3   | 4   | 5   | 8   | 8   |
| Funcionário 1 | 7   | 3   | 4   | 3   | 3   | 4   | 4   |
| Funcionário 2 | 3   | 3   | 4   | 3   | 3   | 2   | 2   |
| Funcionário 3 | 9   | 3   | 4   | 7   | 3   | 4   | 1   |

Escreva um programa (classe `DiaSemana`) que leia uma matriz  $N \times 7$ , sendo  $N > 1$ , e imprima qual(is) o(s) dia(s) da semana em que os funcionários mais trabalham. A saída do programa deverá ser um número inteiro de 1 a 7, indicando respectivamente domingo, segunda, ..., sábado. Entretanto, se houver coincidência de dois ou mais dias serem igualmente os de mais horas trabalhadas, os números correspondentes a todos eles devem ser impressos (um em cada linha).

Considere o número -1 como o final da entrada, ou seja, os 7 primeiros números correspondem ao `Funcionário 0`, seguido dos 7 números para o `Funcionário 1`, e assim por diante.

### Exemplos de entrada e saída esperada:

- Entrada: 2 4 3 4 5 8 8 7 3 4 3 3 4 4 3 3 4 3 3 2 2 9 3 4 7 3 4 1 -1  
Saída: 1
- Entrada: 2 2 3 4 5 8 8 7 7 4 3 3 4 4 3 3 4 3 3 2 2 9 9 4 7 3 4 1 -1  
Saída: 1  
2

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "DiaSemana.java" .....

## Questão 39 – Distância Percorrida pelo Avião

0.00 / 0.40

O tempo que um determinado avião dispensa para percorrer o trecho entre duas cidades distintas está disponível através da seguinte tabela:

|     | 111 | 222 | 333 | 444 | 555 | 666 | 777 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 111 | 0   | 2   | 11  | 6   | 15  | 11  | 1   |
| 222 | 2   | 0   | 7   | 12  | 4   | 2   | 15  |
| 333 | 11  | 7   | 0   | 11  | 8   | 3   | 13  |
| 444 | 6   | 12  | 11  | 0   | 10  | 2   | 1   |
| 555 | 15  | 4   | 8   | 10  | 0   | 5   | 13  |

|     |    |    |    |   |    |    |    |
|-----|----|----|----|---|----|----|----|
| 666 | 11 | 2  | 3  | 2 | 5  | 0  | 14 |
| 777 | 1  | 15 | 13 | 1 | 13 | 14 | 0  |

Escreva um programa que, tendo a tabela acima armazenada como uma matriz, leia uma sequência de números correspondentes às cidades dessa tabela até que o número **-1** seja lido. Em seguida, mostre ao usuário o tempo necessário para percorrer o circuito de cidades por ele informadas.

#### Exemplos de entrada e saída esperada:

- Entrada: 222 444 333 555 -1  
Saída: 31
- Entrada: 777 111 777 -1  
Saída: 2

#### Dica:

- Em Java, você pode inicializar um vetor ou matriz diretamente com seus dados. Exemplo de inicialização de um vetor: `int[] code = { 4, 5, 1 };`

O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.

Enviar "DistanciaAviao.java" .....

## Questão 40 – Fração Contínua: Raiz Quadrada de Dois

0.00 / 0.40 

### Contextualização

Seja  $\mathbb{Z}$  o conjunto dos números inteiros. Sabemos que a quantidade de elementos em  $\mathbb{Z}$  é infinita. Contudo, tomando-se dois números inteiros  $z_1$  e  $z_2$  quaisquer, com  $z_1 \leq z_2$ , teremos uma quantidade *finita* de elementos  $z_i$  tais que  $z_1 \leq z_i \leq z_2$ , ou seja, compreendidos entre  $z_1$  e  $z_2$ . Por outro lado, no conjunto  $\mathbb{R}$  dos números reais, tomando-se dois elementos  $r_1$  e  $r_2$  quaisquer, com  $r_1 \leq r_2$ , teremos uma quantidade *infinita* de elementos  $r_i$  tais que  $r_1 \leq r_i \leq r_2$ . Esse fato dificulta a representação de números reais em máquinas com recursos limitados de armazenamento, tais como os computadores. Uma vez que a memória de qualquer computador é finita, não há como representar nela todos os números reais, cuja quantidade é infinita.

Uma maneira de contornar esse problema é utilizar *frações contínuas* (assunto desta questão) para representar números reais, pois elas nos fornecem aproximações racionais surpreendentemente boas de números reais, além de serem conceitualmente simples. De um modo geral, uma fração contínua é uma expressão da seguinte forma:

$$a_0 + \frac{b_1}{a_1 + \frac{b_2}{a_2 + \frac{b_3}{a_3 + \dots}}}$$

Note que a expressão acima também é infinita, tal como uma dízima periódica.

Quando usamos um algoritmo para computar uma *série infinita*, como a série  $S$  abaixo, nós conhecemos o primeiro termo (neste caso, é o termo 1), mas não conhecemos o último termo.

$$S = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \dots$$

Por outro lado, quando usamos um algoritmo .....para computar uma *fração contínua*, como no exemplo abaixo, nós não conhecemos o primeiro termo, pois ele remonta ao infinito (simbolizado pelo  $\dots$ ):

$$\sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

Por conta disso, ao tratar frações contínuas de forma algorítmica, somos obrigados a arbitrar um começo qualquer, o qual chamamos de "*semente*". Embora a escolha da semente seja arbitrária, o que influencia a qualidade da aproximação é o número de passos (divisões) que resolvemos a fração contínua. Escolhamos apenas um passo:

$$\sqrt{2} \approx 1 + \frac{1}{2 + \frac{1}{semente}}$$

Por exemplo, se arbitramos que semente=10, o resultado aproximado da fração acima será o seguinte:

$$\sqrt{2} \approx 1 + \frac{1}{2 + \frac{1}{10}} = 1 + \frac{1}{\frac{21}{10}} = 1 + \frac{10}{21} = \frac{31}{21} \approx 1.4762$$

distante do valor esperado, que é 1.4142. Contudo, usando esse mesmo valor de semente na aproximação abaixo,

$$\sqrt{2} \approx 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{\text{semente}}}}}}}$$

o resultado será **1.4143**, que é bem mais próximo do valor esperado.

Além disso, note que a fração contínua de  $\sqrt{2}$  é composta por duas partes:

1. Constante: o número **1** logo após o sinal de "aproximado".
2. Cíclica:  $1/(2 + \dots)$

### Problema

Escreva um programa (classe **RaizDois**) que leia um número **N** do teclado e determine o valor de  $\sqrt{2}$  através da sua expansão em fração contínua com **N** termos. Na saída, devem ser mostrados os resultados intermediários, com 14 casas decimais de precisão. Utilize uma semente igual a **1**. **Para esta questão, use apenas variáveis do tipo **double**.**

$$\sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \dots}}}$$

### Dicas:

- Use apenas variáveis do tipo **double** para não perder precisão e não dar erro na correção automática.
- Na expressão da fração contínua, identifique qual a parte constante e qual a parte que se repete. Esta última deverá ser controlada por uma variável acumuladora dentro de um laço de repetição.
- Utilize uma variável acumuladora para guardar o valor parcial da parte cíclica, e uma variável contadora para controlar o número de iterações. Note que elas são independentes uma da outra.
- Para ter 14 casas decimais, o uso do **double** é obrigatório, uma vez que o **float** não tem essa precisão.

### Exemplos de entrada e saída esperada:

- Entrada: **1**  
Saída: **1.33333333333333**
- Entrada: **2**  
Saída: **1.....33333333333333**  
**1.42857142857143**
- Entrada: **5**

|        |                   |
|--------|-------------------|
| Saída: | 1.333333333333333 |
|        | 1.42857142857143  |
|        | 1.41176470588235  |
|        | 1.41463414634146  |
|        | 1.41414141414141  |

---

*O prazo de entrega do trabalho terminou. Portanto, o botão abaixo está desabilitado.*