



ALGORITMOS E PROGRAMAÇÃO I



01 |

Implemente um programa em Java que **informe o tipo do triângulo** a partir do tamanho de seus três lados. O triângulo pode ser Isósceles, Escaleno ou Equilátero.

** Você deve pesquisar sobre como identificar seu tipo.*

Entrada	Saída
6 6 6	"Esse é um triângulo Equilátero"
4 5 6	"Esse é um triângulo Escaleno"

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Tipo Triângulo

Nomeação

```
tipoTriangulo(l1, l2, l3)
```

Chamada

```
tipoTriangulo(6, 6, 6)  
= "Equilátero"
```



Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

02 |

Implementar um programa em Java que **informe a cor resultante** a partir de duas cores primárias. Se as cores informadas não forem primárias, a função deve informar uma mensagem dizendo que apenas cores primárias são aceitas.

Entrada	Saída
"Vermelho" "Azul"	"A combinação das cores é Roxo"
"Vermelho" "Rosa"	"Apenas cores primárias são aceitas"

Função Combinar Cores Primárias

Nomeação

```
combinarCores(cor1, cor2)
```

Chamada

```
combinarCores("Vermelho", "Azul")  
= "Roxo"
```



03 |

Implemente um programa em Java que **informe a classificação de um aluno** a partir de 3 notas e da quantidade de faltas. O aluno já será considerado reprovado se a quantidade de faltas for maior que 30. A tabela abaixo apresenta as regras para os intervalos da média:

Média	Situação
Maior igual a 8	Aprovado com Sucesso
Maior igual a 6 e menor que 8	Aprovado
Maior igual a 3 e menor que 6	Recuperação
Menor que 3	Reprovado
Igual a 0	Desistente

Entrada	Saída
8.0 8.0 8.0 31	"Situação do Aluno: Reprovado"

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Classificação de Aluno

Nomeação

```
situacao(n1, n2, n3, faltas)
```

Chamada

```
situacao(8.0, 8.0, 8.0, 31)  
= "Reprovado"
```



04 |

Implemente um programa em Java que **represente uma calculadora**. A função deve receber dois valores numéricos e o tipo da operação matemática que deve realizar nos valores e retornar o resultado do cálculo. As operações suportadas estão listadas abaixo, caso seja enviada uma operação diferente, um erro deve ser lançado com a mensagem: ***"Operação não suportada"***.

Operações
Soma
Subtração
Multiplicação
Divisão
Resto da Divisão
Potência

Entrada	Saída
2.0 3.0 "Potência"	"Resultado é 8.0"
2.0 3.0 "Raiz"	"Operação não suportada"

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Calculadora

Nomeação

```
calculadora(num1, num2, operacao)
```

Chamada

```
calculadora(2.0, 3.0, "Potência")  
= 8.0
```



Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

05 |

Implemente um programa em Java que **informe qual a situação** de um candidato de vestibular a partir da nota que ele tirou, da nota de corte e da nota mínima para aprovação. Considere as regras abaixo:

Média	Situação
Se a nota for menor que a nota de corte	Candidato não passou
Se a nota for maior ou igual a nota de aprovação	Candidato aprovado
Se a nota estiver entre a nota de corte e a nota mínima de aprovação	Candidato está na lista de espera

Entrada	Saída
8.1 7.5 8.0	"Candidato aprovado"

Função Situação Candidato

Nomeação

```
situacao(nota, corte, minima)
```

Chamada

```
situacao(8.1, 7.5, 8.0)  
= "Candidato aprovado"
```



06 |

Implemente um programa em Java que **calcule o valor da mensalidade** de um aluno de faculdade de tecnologia, a partir da sigla do curso, se é isento (sim ou não) e do desconto em (%) que ele possui. Ao lado está o valor da mensalidade para cada curso:

Curso	Mensalidade
SI – Sistemas de Informação	R\$ 550,00
ADS – Análise e Desenv. Sistemas	R\$ 750,00
CS – Ciência da Computação	R\$ 1.150,00
EC – Engenharia da Computação	R\$ 1.300,00
ES – Engenharia de Software	R\$ 950,00

Entrada	Saída
"SI" false 0	550.00
"ADS" false 50	375.00
"CS" true 0	0.00

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Mensalidade Aluno

Nomeação

```
mensalidade(  
    sigla,  
    isento,  
    desconto)
```

Chamada

```
mensalidade("SI", false, 0.0)  
= 550.0
```




07 |

Implemente um programa em Java que **calcule o total a se pagar** na compra de ingressos de cinema, a partir da quantidade de ingressos Inteiras e Meias, do dia da semana que será passado o filme e se o filme é nacional. O valor do ingresso é R\$ 28,50. Considere que:

- Se o ingresso for para quarta-feira, o preço do ingresso é único R\$ 14,25.
- Se for um filme nacional, o preço do ingresso é único R\$ 5,00.
- O desconto para filmes nacionais tem prioridade sobre o desconto de quarta-feira, ou seja, os dois descontos não podem acontecer ao mesmo tempo.

Entrada	Saída
1 2 "quarta-feira" true	O total a se pagar pelos ingressos é R\$ 5.0

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Total Compra Ingressos

Nomeação

```
totalCompra(  
    inteiras,  
    meias,  
    diaSemana,  
    nacional)
```

Chamada

```
totalCompra(  
    1,  
    2,  
    "quarta-feira",  
    true) = 5.0
```



08 |

Implemente um programa em Java que informe a **situação de um orçamento familiar** baseado no total de ganhos e gastos. A situação deve ser calculada a partir da tabela abaixo:

Média	Situação
Gastos menores que 30% dos Ganhos	Parabéns, está gerenciando bem seu orçamento!
Gastos menores que 50% dos Ganhos	Muito bem, seus gastos não ultrapassam metade dos ganhos!
Gastos menores que 80% dos Ganhos	Atenção, melhor conter os gastos!
Gastos menores que 100% dos Ganhos	Cuidado, seu orçamento pode ficar comprometido!
Gastos maiores que os Ganhos	Orçamento comprometido! Hora de rever seus gastos!

Entrada	Saída
3000.0 2700.0	"Atenção, melhor conter os gastos!"
3200.0 2700.0	"Orçamento comprometido! Hora de rever seus gastos!"

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Situação Orçamento Familiar

Nomeação

```
sitOrçamento(ganhos, gastos)
```

Chamada

```
sitOrçamento(3000.0, 2700.0)  
= "Atenção, melhor conter os gastos!"
```



09 |

Implemente um programa em Java que simule o **jogo "Par ou Ímpar"**. O usuário deve informar sua jogada e escolher se quer "Par" ou "Ímpar", enquanto seu oponente será a máquina, que realizará uma jogada aleatória de 1 a 5. Ao final, o programa deve apresentar as jogadas feitas e o vencedor.

Entrada	Saída
"Par" 2	Jogador pediu Par Jogador 2 x 4 Máquina Jogador venceu!
"Par" 2	Jogador pediu Par Jogador 2 x 5 Máquina Máquina venceu!

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Jogada Máquina

Nomeação

```
jogadaBot()
```

Chamada

```
jogadaBot() = 4
```

Função Ganhador Par ou Ímpar

Nomeação

```
parOuImpar(  
    jogadorEscolha,  
    jogador,  
    bot)
```

Chamada

```
parOuImpar("Par", 2, 4)  
= "Jogador venceu"
```



10 |

Implemente um programa em Java que represente a lógica do **jogo Jokenpô**. O usuário deve informar qual sua jogada: Pedra, Papel ou Tesoura, enquanto o oponente será a máquina que realizará uma jogada aleatória. Ao final, o programa deve apresentar as jogadas feitas e o vencedor.

Obs.: Para a jogada aleatória utilize a função ***Math.random()*** para gerar um número aleatório de 1 a 3. Considere: **1 -> Pedra** **2 -> Papel** **3 -> Tesoura**.

Entrada	Saída
"Papel"	Joo-kenn-poo! Jogador Papel x Pedra Máquina Jogador venceu!

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Jogada Máquina

Nomeação

jogadaBot()

Chamada

jogadaBot() = "Pedra"

Função Ganhador Jokenpo

Nomeação

jokenpo(
 jogador,
 bot)

Chamada

jokenpo("Papel", "Pedra")
 = "Jogador venceu"



Bons estudos!
Bruno de Oliveira