Bruno de Oliveira

São Paulo

ADO

Exercícios

ALGORITMOS E PROGRAMAÇÃO I

Bruno de Oliveira

São Paulo

TADS

01 |

Implemente um programa em Java que a partir de **dois números** informados pelo usuário, **calcule o dobro de cada número** e apresente ao usuário.

***Obs:** Lembre que uma função só pode retornar 01 valor, mas você pode chamar a mesma função passando valores diferentes.*

Entrada	Saída
8 6	O dobro de 8 é 16 O dobro de 6 é 12
4 11	O dobro de 4 é 8 O dobro de 11 é 22

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

Função Dobro

Nomeação

```
dobro(double numero) => double
```

Chamada

```
dobro(8) = 16
```

02 |

Implemente um programa em Java que a partir de três notas informados pelo usuário, **calcule a média** e **verifique se o aluno passou,** sabendo que a média para passar é 6.0. Ao final apresente ao usuário a média e se o aluno passou na disciplina.

***Obs:** Assim que você guarda a resposta de uma função, você pode usar essa resposta para ser enviada para outra função. No caso ao lado, após chamar a função media, você deve usar sua resposta para chamar a função passou.*

Entrada	Saída
8 6 4	A média é 6.0 Aluno passou? true
5.5 6.5 6	A média é 6.0 Aluno passou? true

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

Função Média

Nomeação

```
media(double nota1,
      double nota2,
      double nota3) => double
```

Chamada

```
media(5.5, 6.5, 6) = 6.0
```

Função Passou

Nomeação

```
passou(double media) => boolean
```

Chamada

```
passou(6.0) = true
```

03 |

Implementar um programa em Java para **verificar** se dois retângulos possuem a mesma área.

***Obs:** Uma função pode chamar outra função em seu código. No caso ao lado a função iguais deve chamar a função área duas vezes para depois comparar os valores.*

Entrada	Saída
4 2 3 4	“Retângulos são iguais? false”
6 2 3 4	“Retângulos são iguais? true”

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

Função Área do Retângulo

Nomeação

```
area(double base, double altura)
  => double
```

Chamada

```
area(4, 2) = 8
```

Função Retângulos Iguais

Nomeação

```
iguais(double b1,
       double a1,
       double b2,
       double a2) => boolean
```

Chamada

```
iguais(4, 2, 3, 4) = false
```

04 |

Implementar um programa em Java para **verificar** se duas cores são primárias.

Entrada	Saída
azul amarelo	“As duas cores são primárias? true”
azul roxo	“As duas cores são primárias? false”

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

Função Cor é Primária

Nomeação

```
primaria(String cor) => boolean
```

Chamada

```
primaria("azul") = true
```

Função Duas cores Primárias

Nomeação

```
primarias(String cor1, String cor2) => boolean
```

Chamada

```
primarias("azul", "roxo") = false
```

05 |

Implementar um programa em Java para **verificar** se uma senha é forte**.** Para ser forte ela precisa ter no mínimo 6 caracteres, um número e um caractere especial.

Caracteres especiais: ! @ # $ % ^ & * ( ) _

Obs: *Veja que na organização de funções ao lado, a função senhaForte deve chamar as outras três funções para depois verificar se todas elas responderam verdadeiro.*

Entrada	Saída
"admin@123"	"A senha cumpre os requisitos? true"
“admin@!#"	"A senha cumpre os requisitos? false"

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

Função Possui Caractere Especial

Nomeação
```
possuiCaractereEspecial(String senha) => boolean
```
Exemplo de Chamada
```
possuiCaractereEspecial("admin123") = false
```

Função Possui Número

Nomeação
```
possuiNumero(String senha) => boolean
```
Exemplo de Chamada
```
possuiNumero("admin123") = true
```

Função Cumprimento Válido

Nomeação
```
cumprimentoValido(String senha) => boolean
```
Exemplo de Chamada
```
cumprimentoValido("admin123") = true
```

Função Senha é Forte

Nomeação
```
senhaForte(String senha) => boolean
```
Exemplo de Chamada
```
senhaForte("admin123") = false
```

Bons estudos!

Bruno de Oliveira