



ALGORITMOS E PROGRAMAÇÃO I



01 |

Implemente um programa em Java que peça ao usuário 05 números inteiros e armazene-os em um **Array**. Após isso, o programa deve **escrever os números no terminal**.

Entrada	Saída
1 2 3 4 5	Você digitou os números: 1 2 3 4 5
10 20 30 40 50	Você digitou os números: 10 20 30 40 50

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

Função Exibir Números

Definição

```
exibir(numeros: int[]) : void
```

Exemplo Chamada

```
exibir(new int[] { 1, 2, 3, 4, 5 })
```

Terminal

```
1  
2  
3  
4  
5
```



02 |

Implemente um programa em Java que peça ao usuário 05 números decimais e armazene-os em um **Array**. Após isso, o programa deve **escrever os números no terminal**.

Entrada	Saída
1.0 2.5 3.8 4.2 5.3	Você digitou os números: 1.0 2.5 3.8 4.2 5.3

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

Função Exibir Números

Definição

```
exibir(numeros: double[]) : void
```

Exemplo Chamada

```
exibir(new double[] { 1.0, 2.5, 3.8,  
                     4.2, 5.3 })
```

Terminal

```
1.0  
2.5  
3.8  
4.2  
5.3
```



03 |

Implemente um programa em Java que peça ao usuário 05 nomes e armazene-os em um **Array**. Após isso, o programa deve **escrever os nome no terminal**.

Entrada	Saída
Bruno Ingrid Luiza Junior Lucia	Você digitou os nomes: Bruno Ingrid Luiza Junior Lucia

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

Função Exibir Números

Definição

```
exibir(nomes: String[]) : void
```

Exemplo Chamada

```
exibir(new String[] {  
    "Bruno",  
    "Ingrid",  
    "Luiza",  
    "Junior",  
    "Lucia"  
})
```

Terminal

```
Bruno  
Ingrid  
Luiza  
Junior  
Lucia
```



04 |

Implemente um programa em Java que inicialmente peça ao usuário a quantidade de números a serem armazenados em um **Array**. Em seguida, peça ao usuário que informe um valor para cada posição. Ao final, **escreva os números informados no terminal**.

Entrada	Saída
7	Você digitou os números:
10	10
20	20
30	30
40	40
50	50
60	60
70	70

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

Função Criar Array

Definição

```
criarArray(tamanho: int) : int[]
```

Objetivo

Criar um array com o tamanho recebido, e preenchê-lo com valores informados pelo usuário. Essa função terá interação com o usuário.

Função Criar Array

Definição

```
exibir(numeros: int[]) : void
```

Objetivo

Escreve os números do array no terminal.



05 |

Implemente um programa em Java que inicialmente peça ao usuário a quantidade de números a serem armazenados em um **Array**. Em seguida, peça ao usuário que informe um valor para cada posição. Com o array preenchido, o programa deve **calcular o dobro de cada elemento** e armazenar em outro array. Ao final, apresente o array inicial, e o array com os dobros.

Entrada	Saída
5	Array inicial:
10	10 20 30 40 50
20	
30	Array com Dobros:
40	20 40 60 80 100
50	

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

Definição

```
criarArray(tamanho: int) : int[]
```

Objetivo

Criar um array com o tamanho recebido, e preenchê-lo com valores informados pelo usuário. Essa função terá interação com o usuário.

Definição

```
calcularDobros(numeros: int[]) : int[]
```

Objetivo

Retorna um novo array com o dobro de cada elemento.

Definição

```
exibir(numeros: int[]) : void
```

Objetivo

Escreve os números do array no terminal.



06 |

Implemente um programa em Java que **calcule a tabuada de um número**, a partir de um número informado pelo usuário. O programa deve guardar cada número da tabuada em um **Array**. Ao final, apresente o array conforme abaixo.

Entrada	Saída
5	A tabuada do 5 é: 0 5 10 15 20 25 30 35 40 45 50

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

Definição

```
tabuada(numero: int) : int[]
```

Objetivo

Retorna um novo array contendo a tabuada de um número informado.

Definição

```
exibir(numeros: int[]) : void
```

Objetivo

Escreve a tabuada no formato correto



07 |

Implemente um programa em Java que inicialmente peça ao usuário a quantidade de notas a serem armazenadas em um **Array**. Em seguida, peça ao usuário que informe cada uma das notas. Com o array preenchido, o programa deve **calcular a Média, a Maior e a Menor nota**. Ao final, apresente os dados conforme abaixo

Entrada	Saída
5	A maior nota é 8.0
3.5	A menor nota é 2.0
2.0	A média é 5.0
6.5	
8.0	Notas digitadas:
5.0	3.5 - 2.0 - 6.5 - 8.0 - 5.0

Definição

```
criarArray(tamanho: int) : double[]
```

Objetivo

Criar um array com o tamanho recebido, e preenchê-lo com valores informados pelo usuário. Essa função terá interação com o usuário.

Definição

```
media(notas: double[]) : double
```

Objetivo

Retorna a média das notas.

Definição

```
maior(notas: double[]) : double
```

Objetivo

Retorna a maior nota.

Definição

```
menor(notas: double[]) : double
```

Objetivo

Retorna a menor nota.

Definição

```
exibir(notas: double[]) : void
```

Objetivo

Escreve as notas no formato correto



08 |

Implemente um programa em Java que inicialmente peça ao usuário a quantidade de números a serem gerados **automaticamente entre 0-10** em um **Array**. Em seguida, peça ao usuário que informe **um número** para que o programa verifique se **existe dentro do Array**. O programa deve retornar a posição do número no array caso o encontre, ou retornar “não encontrado”.

Entrada	Saída
10 5	O número 5 encontra-se na posição 3. Números gerados: 3 6 2 5 7 6 4 2 0 1
10 5	O número 5 não foi encontrado. Números gerados: 3 6 2 4 7 6 4 2 0 1

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

Definição

```
criarArray(tamanho: int) : int[]
```

Objetivo

Cria um array com o tamanho recebido preenchido com números aleatórios entre 0-10.

Definição

```
buscar(num: int, numeros: int[]) : int
```

Objetivo

Retorna a posição do número no array ou -1 se não encontrar.

Definição

```
exibir(numeros: int[]) : void
```

Objetivo

Escreve os números no formato correto.



09 |

Implemente um programa em Java que **conte a quantidade de palavras** contidas em uma frase informada pelo usuário. O programa deve apresentar a quantidade de palavras e lista-las conforme abaixo:

Entrada	Saída
Bora codar para conseguir um estágio!	A frase contém 6 palavras, são elas: Bora codar para conseguir um estágio!

Considere a função `split(...)` para transformar uma `String` em `Array` a partir de um critério:

```
String frase = "Sangue nos Oio!";  
String[] palavras = frase.split(" ");
```

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

Definição

```
mapearPalavras(frase: String) : String[]
```

Objetivo

Retorna um array contendo as palavras contidas em uma frase.

Definição

```
exibir(palavras: String[]) : void
```

Objetivo

Escrever as Strings do array no terminal.



10 |

Implemente um programa em Java para ajudar em um debate que acontecerá em uma sala de aula. O programa deve **sortear os alunos aleatoriamente para formar dois grupos**. Para isso, cada aluno recebeu um número sequencial começando em 01. Deve ser informado ao programa, a quantidade de alunos e o programa deve criar dois arrays com os alunos sorteados aleatoriamente. O programa deve garantir que não terá alunos repetidos nos grupos. Ao final, apresente os dois arrays.

Entrada	Saída
20	Primeiro grupo: 3 - 6 - 12 - 5 - 17 - 8 - 19 - 1 - 14 - 10 Segundo grupo: 2 - 4 - 7 - 20 - 11 - 9 - 15 - 18 - 13 - 16

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

Definição

```
buscar(num: int, numeros: int[]) : int
```

Objetivo

Retorna a posição do número no array ou -1 se não encontrar.

Definição

```
gerarNovo(tamSala: int,  
          grupo1: int[],  
          grupo2: int[]) : int
```

Objetivo

Gera um número aleatório até que não esteja em nenhum dos grupos.



Bons estudos!
Bruno de Oliveira