



ALGORITMOS E PROGRAMAÇÃO I



01 |

Implemente um programa em Java que **simule uma lista de tarefas**, o programa deve continuar pedindo uma tarefa ao usuário até que o usuário digite **“Sair”**. Após isso, o programa deve **escrever as tarefas no terminal**.

Entrada	Saída
Estudar cedo Trabalhar até tarde Preparar presente para o prof Iniciar um livro de programação Sair	Suas tarefas do dia são: 1. Estudar cedo 2. Trabalhar até tarde 3. Preparar presente para o prof 4. Iniciar um livro de programação
Dormir o dia todo Sair	Suas tarefas do dia são: 1. Dormir o dia todo

Função Adicionar Item ao Array

Definição

```
adicionar(tarefas: String[],  
          novaTarefa: String) : String[]
```

Objetivo

Criar um novo array com as tarefas já existentes e adiciona a nova tarefa.

Função Exibir Números

Definição

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

Exemplo Chamada

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

Terminal

```
Bruno  
Ingrid  
Luiza  
Junior  
Lucia
```



02 |

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. Após isso, o programa deve **inverter os números no array**. **Utilize apenas um array.*

Entrada	Saída
5 10 25 20 40 30	Os números invertidos são: 30 40 20 25 10
4 10 30 40 20	Os números invertidos são: 20 40 30 10

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

Função Inverter Array

Definição

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

Objetivo

Inverte os números do array recebido e retorna o mesmo array.

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 Exibir Números

Definição

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

Objetivo

Escrever no terminal cada item do array.



03 |

Implemente um programa em Java que gere **automaticamente um jogo** para uma loteria federal. **O jogo deve ter 15 números entre 01 e 25.** Você deve garantir que os números do jogo gerado não se repitam. Ao final, os números devem ser apresentados ao usuário.

Entrada	Saída
	9 – 2 – 16 – 13 – 15 4 – 7 – 17 – 18 – 11 1 – 6 – 21 – 22 – 24

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

Função Gerar Novo Número

Definição

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

Objetivo

Repete o processo de gerar um número aleatório entre 1 e 25, enquanto ele existir no array recebido.

Função Exibir Números

Definição

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

Objetivo

Escrever no terminal cada item do array.



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. Após isso, o programa deve **verificar se os números estão em Ordem crescente, Ordem decrescente ou Desordenados**.

Entrada	Saída
3 10 11 12	Organização dos elementos: Ordem crescente
3 12 11 10	Organização dos elementos: Ordem decrescente
3 10 12 11	Organização dos elementos: Desordenados

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

Função Verificar Ordem

Definição

```
verificarOrdem(numeros: int[]) : String
```

Objetivo

Verifica se o array está em ordem Crescente, Decrescente ou Desordenados.

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.



05 |

Implemente um programa em Java que peça ao usuário um número que será usado para gerar o tamanho de um Array contendo a **sequência dos números Primos**. O tamanho da sequência é o número informado pelo usuário.

Entrada	Saída
5	Sequência dos 5 primeiros primos: 2 3 5 7 11
7	Sequência dos 7 primeiros primos: 2 3 5 7 11 13 17

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

Função Sequência Primos

Definição

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

Objetivo

Cria um array com a sequência de números primos com o tamanho recebido.

Função Exibir Números

Definição

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

Objetivo

Escrever no terminal cada item do array.



06 |

Implemente um programa em Java que peça ao usuário um número que será usado para gerar o tamanho de um Array contendo a **sequência de Fibonacci**. O tamanho da sequência é o número informado pelo usuário.

Entrada	Saída
5	Sequência de Fibonacci com 5 números: 1 1 2 3 5
7	Sequência de Fibonacci com 7 números: 1 1 2 3 5 8 13

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

Função Sequência Fibonacci

Definição

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

Objetivo

Cria um array com a sequência de Fibonacci com o tamanho recebido.

Função Exibir Números

Definição

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

Objetivo

Escrever no terminal cada item do array.

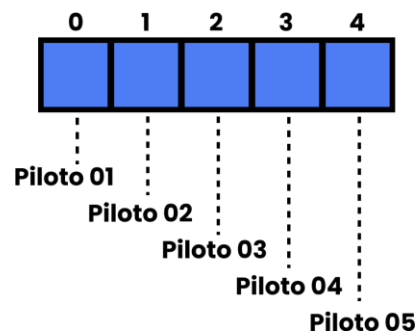


07 |

Implemente um programa em Java para identificar **a performance de cada piloto da Fórmula 01** em uma corrida oficial. Inicialmente o programa deve pedir a quantidade de pilotos competindo. O usuário deve informar a posição de cada piloto no **momento da largada** e armazená-las em um Array. Depois, o usuário deve informar a posição dos pilotos no **momento da chegada** e guardá-las em outro Array. Para cada piloto, o programa deve apresentar sua performance dizendo quantas posições ele avançou, retrocedeu ou se manteve a posição.

Entrada	Saída
5	O piloto 1 avançou 2 posições
1	O piloto 2 retrocedeu 2 posições
4	O piloto 3 avançou 2 posições
2	O piloto 4 retrocedeu 2 posições
3	O piloto 5 manteve a posição
5	
3	
2	
4	
1	
5	

* O número do piloto é referente a posição do array.



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

Função Verificar Ordem

Definição

```
performancePiloto(numPiloto: int,  
posLargada: int,  
posChegada: int) : String
```

Objetivo

Verifica a performance do piloto a partir de sua posição de largada e chegada.

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.



08 |

Implemente um programa em Java que ajuda os alunos de uma turma a montar uma **fila Indiana por ordem de tamanho**. Inicialmente peça ao usuário a quantidade de alunos de uma sala de aula. Depois disso, o usuário deve informar a altura de cada aluno para serem armazenadas em um Array. O programa deve ordenar o Array e apresentar ao usuário.

Entrada	Saída
5	Fila indiana:
1.55	1.47
1.47	1.49
1.61	1.55
1.49	1.60
1.60	1.61

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

Função Ordenar

Definição

```
ordenar(alturas: double[]) : double[]
```

Objetivo

Ordenar as alturas recebidas no mesmo array e retorná-lo.

Função Criar Array

Definição

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

Objetivo

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

Função Exibir Números

Definição

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

Objetivo

Escrever no terminal cada item do array.



09 |

Implemente um programa em Java que ajude a avaliar **quantas trocas de figurinhas** da copa do mundo, podem ser feitas entre duas pessoas. O programa deve ler os números das figurinhas da primeira pessoa e armazenar em um Array. O mesmo deve ser feito para a segunda pessoa. O programa deve **pedir a quantidade de figurinhas que serão informadas antes de ler as figurinhas**. Ao final, o programa deve apresentar quantas trocas podem ser feitas considerando que qualquer troca é válida.

Entrada	Saída
5 10 20 100 110 120	Podem ser feitas 2 trocas.
4	
10	
20	
200	
210	

Função Figurinhas em Comum

Definição

```
figurinhasIguais(pessoa1: int[],  
                 pessoa2: int[]) : int[]
```

Objetivo

Retorna um array com os números das figurinhas em comum das duas pessoas.

Função Trocas

Definição

```
qtdTrocas(pessoa1: int[],  
          pessoa2: int[],  
          emComum: int[]) : int
```

Objetivo

Retorna a quantidade de trocas que podem ser feitas entre as pessoas.

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.



10 |

Implemente um programa em Java que simule um software **para analisar se duas pessoas são compatíveis em uma rede social, a partir de seus interesses**. O programa deve pedir 5 interesses de cada pessoa e armazená-los em dois **Array's** diferentes. Ao final, o programa deve mostrar a porcentagem de similaridade de interesses, se for maior que 50%, o programa deve apresentar: **"Deu Match!"**, senão, **"Fila anda"**.

Entrada	Saída
Filmes Séries Esportes Viagens Futebol Jogos Trilha Séries Filmes Esportes	Similaridade: 60% Deu Match!

Função Trocas

Definição

```
grauSimilaridade(pessoa1: String[],  
                 pessoa2: String[],  
                 emComum: String[]) : String
```

Objetivo

Retorna uma mensagem contendo a porcentagem da similaridade e o resultado da compatibilidade: "Deu Match" ou "Fila Anda".

Função Interesses Similares

Definição

```
similares(pessoa1: String[],  
          pessoa2: String[]) : String[]
```

Objetivo

Retorna um array com os interesses em comum das duas pessoas.

Função Criar Array

Definição

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

Objetivo

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



Bons estudos!
Bruno de Oliveira