

EXERCÍCIOS

1. Carregar um vetor [100] real de valores de saldos. Calcular e exibir:

- Média dos saldos positivos entre 100 à 1000
- Média geral dos saldos
- Soma dos saldos negativos

2. Carregar um vetor [5] inteiro. Enviar cada elemento para uma função e esta irá retornar o seu fatorial que será armazenado em um outro vetor. Exibir os dados dos vetores

3. Carregar um vetor [100] inteiros positivos ou negativos. Classificar este vetor em ordem crescente e apresentar os valores.

4. Carregar uma matriz [4 x 4 inteiro]. Apresentar:

- Soma dos valores no intervalo de 1 a 100
- Quantidade de números ímpares entre 30 a 50
- Quantidade de números divisíveis por 8
- Quantidade de números ímpares divisíveis por 3
- Fatorial do maior número informado na matriz

5. Criar e carregar uma matriz [4 x 4] inteiro, onde os valores da diagonal principal serão carregados pela aplicação conforme o gráfico e os demais dados serão digitados pelo usuário

1			
	3		
		9	
			27

6) Simule o método a seguir, demonstrando cada passo em uma representação gráfica de lista:

```
public void metodoA (int pos, int valor)
```

```
{
    if (dados.length == tamanho)

        System.out.println("ERRO!");
    else { if (pos <= 0 || pos > tamanho+1)

        System.out.println("Posição Inválida!");

    else {
        if (pos == 1)

            adicionaInicio(valor);
        else
            if (pos == tamanho)

                adicionaFinal(valor);
            else{
                for (int i = tamanho; i >= pos; i--)
```

```

        dados[i]=dados[i-1];
        dados[pos-1]=valor;
        tamanho++;
    }
}
}
}

```

7) Implemente na classe Lista um método que remove o elemento de uma posição específica recebida por parâmetro.

8) Implemente uma Lista de caracteres em alocação estática com todas as operações indicadas a seguir:

-verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;

-verificar se a lista está cheia, retornando true se estiver cheia e false se não estiver;

-adicionar caractere no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar caractere no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar caractere em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando o motivo;

-remover caractere do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-remover caractere do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-remover caractere de determinada posição da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-percorrer a lista concatenando os elementos em uma String que será devolvida no final.

9) Implemente uma Lista de Strings em alocação estática com todas as operações indicadas a seguir:

-verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;

-verificar se a lista está cheia, retornando true se estiver cheia e false se não estiver;

-adicionar uma String no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar uma String no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar uma String em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando o motivo;

- remover a String do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a String do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a String de determinada posição da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- percorrer a lista concatenando os elementos em uma String que será devolvida no final.

10) Implemente uma Lista de temperaturas em alocação estática com todas as operações indicadas a seguir:

- verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;
- verificar se a lista está cheia, retornando true se estiver cheia e false se não estiver;
- adicionar uma temperatura no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
- adicionar uma temperatura no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
- adicionar uma temperatura em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando o motivo;
- remover a temperatura do início da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a temperatura do final da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- remover a temperatura de determinada posição da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;
- percorrer a lista concatenando os elementos em uma String que será devolvida no final

11) Elabore um resumo do artigo “A Study on Performance Analysis of Data Structures” disponível no link abaixo:

https://www.academia.edu/37436288/A_Study_on_Performance_Analysis_of_Data_Structures?auto=download

12) Descreva os requisitos funcionais, não funcionais e regras de negócio que pretende aplicar no projeto de Estrutura de Dados e faça um cronograma detalhado de tudo que pretende desenvolver no projeto.

