Universidade Federal de São João del-Rei
Departamento de Ciência da Computação
Estrutura de Dados
Professor Guilherme de Castro Pena

Instruções para entrega:

- Entregue a lista apenas no formato *.pdf* com o nome ***Y_listaX.pdf***, onde **X é o número da lista** e **Y é o número da sua matrícula**. Não serão aceitos outros formatos.
- Inclua nome e matrícula, e mantenha a resolução dos exercícios **ordenada** e **legível**.
- Códigos completos (com int main), em texto, compiláveis, quando aplicável.

  Para cada um, apresente uma imagem da tela de saída do seu programa.
- Após a data de entrega, a nota da entrega é 0.
- Em caso de dúvidas, procurem os monitores. Haverá um monitor após as aulas de laboratório para tirar dúvidas sobre a lista.

Lista de Exercícios 1
Encapsulamento, Análise de Complexidade, TADs, Listas, Pilhas e Filas

Data máxima de entrega: 05/05/2025 - 11:59h
(Entrega: pelo SIGAA, na sua turma de Estrutura de Dados.)

1 Encapsulamento e Tipos Abstrados de Dados (TADs)

Lembrete: A criação de TADs encapsulados é feita pela separação no **.h** (*header*), onde colocam-se a *struct* e as *operações* que o definem. Se desejar manter a implementação das operações também no arquivo **.h** pode.

1.1 Criar um TAD para manipular frações (numerador e denominador inteiros). Defina as seguintes operações:

- Soma, subtração, multiplicação e divisão de duas frações.
- Uma operação que mostra o número decimal da fração.
- Uma operação que simplifique a fração quando possível. Use o MDC.
- Operação para imprimir a fração.
- Outras operações complementares que você achar válidas.

Crie uma função main para testar o seu TAD.

1.2 Crie um TAD: Pedido, que representará um pedido de alimento em um aplicativo de delivery:

Esse TAD é projetado para representar um pedido de comida em um aplicativo de entrega de alimentos, com os seguintes detalhes de implementação:

Atributos:

- itens: Uma lista de itens de comida que o usuário deseja pedir, cada item contendo informações como nome, preço e quantidade.
- endereço_de_entrega: O endereço onde o pedido deve ser entregue.

- total: O custo total do pedido.
- status: O estado atual do pedido (por exemplo, ”pendente”, ”preparando”, ”entregue”).

Métodos:

- adicionar_item(item): Adiciona um item ao pedido, atualizando o total.
- remover_item(item): Remove um item do pedido, atualizando o total.
- calcular_total(): Calcula o custo total do pedido com base nos itens selecionados.
- definir_endereço(endereço): Define o endereço de entrega.
- alterar_status(novo_status): Atualiza o status do pedido (por exemplo, de ”pendente”para ”preparando”).
- finalizar_pedido(): Finaliza o pedido imaginando um retorno para o restaurante (pode apenas imprimir as informações).

Crie uma função main para testar o seu TAD.

2 Listas

2.1 Nas considerações adicionais da aula TAD: Lista Sequencial Estática, é mencionado uma alteração para criar um TAD: Lista Sequencial Dinâmica, onde a característica principal é que o vetor de dados também será alocado de forma dinâmica pelo tamanho max, e a lista ficar cheia, o max pode aumentar +100, e uma nova alocação é feita no vetor de dados.

Implemente esse novo TAD e teste-o. Lembre-se, quando for necessário aumentar a alocação do vetor de dados, deve-se garantir que a cópia dos dados anteriores já inseridos na lista estarão presentes no novo espaço de memória, antes da nova inserção.

2.2 **Desafio:** Na LSE (Lista Simplesmente Encadeada) fizemos uma operação que imprime todos os elementos do início até o fim. Agora faça uma operação que imprima os elementos na ordem contrária, ou seja, do último até o primeiro. Faça a implementação similar à impressão normal, usando repetições, em seguida, pense em uma implementação recursiva, de forma que o comando de impressão, venha após a chamada recursiva do *prox*. Qual delas você achou mais fácil?

2.3 Defina uma struct aluno (nome, matrícula e nota) e crie o TAD: Lista Simplesmente Encadeada para ser uma lista de alunos agora. Mantenha a operação que insere ordenado, ordenando os nomes em forma alfabética, e a remoção de um elemento específico pode usar a matrícula do aluno como chave.

2.4 Repita a mesma operação de imprimir do último até o primeiro agora para a LDE (Lista Duplamente Encadeada), definitivamente o ponteiro *ant* vai ajudar bastante.

2.5 **(Opcional)** Faça os exercícios 5, 8, 9, 15, 20 e 21 do livro do Prof. André Backes (referência no fim dos slides de aula, o livro tem acesso liberado pelo minha biblioteca no pergamum), capítulo 5, a partir da página 191.

3 Pilhas e Filas

3.1 **Beecrowd:** 3 exercícios para trabalhar com Pilhas e Filas estão em uma lista criada no Beecrowd:

- Passo 1: Acesse ou registre na plataforma *Beecrowd* (https://www.beecrowd.com.br/judge/pt/);

- Passo 2: Configure sua universidade ⇒ Entre no menu **PERFIL/CONFIGURAÇÕES/EDUCAÇÃO**. Em **Nome da Instituição**, escreva UFSJ e cadastre suas informações da universidade;
- Passo 3: Cadastre-se na disciplina: ⇒ Entre no menu ACADEMIC e clique em Acessar Disciplina; Na janela que abrir coloque as informações **ID DISCIPLINA:** 11548 e **CHAVE:** J-i3Hfj, e clique em Entrar. Veja a imagem:

- Passo 4: Feito isso, você terá acesso às listas de exercícios;
- Passo 5: Faça primeiro a lista **Hello World!** para relembrar/entender a plataforma. Lembre-se dos problemas de *Presentation Error*, se esquecer um *espaço* ou $\backslash n$ na saída do programa;
- Passo 6: Faça agora a lista **Lista de exercícios 1 - Pilhas e Filas** que contém 3 exercícios que envolvem Pilhas ou Filas, para resolver, use a sua estrutura criada em sala.

3.2 Implemente uma função que receba uma fila e a inverta. Faça a função para ambos os tipos de fila: estática e dinâmica;

3.3 Implemente uma função que receba uma pilha e a inverta. Faça a função para ambos os tipos de pilha: estática e dinâmica;

3.4 Implemente uma função que receba duas filas (F1 e F2) e as concatene. O resultado da concatenação deve ser colocado em F1. A fila F2 deve ficar vazia. Faça a função para ambos os tipos de fila: estática e dinâmica.

3.5 Considerando uma pilha com 4 valores inteiros inseridos na seguinte ordem: 1, 2, 3 e 4; e considerando as operações: insere (I), remove (R) e verTopo (T); Qual a sequência de operações que devemos executar para imprimir os valores de saída na ordem 2 4 3 1?

4 Análise de Complexidade

4.1 Faça a análise de complexidade de pior caso (notação O) de cada uma das operações implementadas em cada um dos TADs (em cada função, identifique a operação básica mais importante para analisar): Lista Sequencial Estática, Lista Simplesmente Encadeada, Lista Duplamente Encadeada, Pilha e Fila.

Não precisa fazer uma análise muito detalhada, apenas olhar para a operação e dizer se é $O(1)$, $O(n)$, $O(n^2)$, etc e justificar de forma breve porque chegou a essa conclusão.