



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Avenida João Naves de Ávila, 2121, Bloco 1B - Bairro Santa Mônica, Uberlândia/MG, CEP 38400-902
Telefone: +55 (34) 3239-4218 - www.facom.ufu.br - cocom@ufu.br



Bacharelado em Ciência da Computação

Bacharelado em Sistemas de Informação

Disciplina: Algoritmos e Estruturas de Dados 1 – AED1 [GBS024/GSI006]

Prof. Me. Claudiney R. Tinoco

Prof. Me. Luiz Fernando Afra Brito

Material baseado: Prof. Dr. Luiz Gustavo Almeida Martins

2º Trabalho de Algoritmos e Estruturas de Dados 1

[15 pontos = 10 de entregáveis + 5 de apresentação]

- Os trabalhos poderão ser feitos em duplas;
- Deverá ser submetido no repositório do GitHub até o dia 08/08/2022;
- A apresentação dos códigos será agendada pelo professor no final da disciplina;
- Os códigos deverão ser implementados somente em Linguagem C, sendo necessária a utilização das estruturas de dados conforme discutidas em sala;
- Cada TAD deverá ser implementando em um projeto diferente;
- O programa cliente deverá possuir um menu interativo para executar as operações.
- Os exercícios marcados como **EXTRA** ficam como sugestão de estudo.

1) Implementar a TAD Pilha usando alocação **estática/sequencial**. Operações que a TAD deve contemplar:

- a) Inicializar_Pilha: (vista em sala)
- b) Pilha_vazia: (vista em sala)
- c) Pilha_cheia: (vista em sala)
- d) Empilha: (vista em sala)
- e) Desempilha: (vista em sala)
- f) Lê_topo: (vista em sala)
- g) Imprimir: imprimir os elementos da pilha, do topo para a base.
- h) [EXTRA] Imprimir_reversa: imprimir os elementos da pilha, da base para o topo.
- i) [EXTRA] Liberar: libera todos os elementos de uma pilha.
- j) Palíndromo: verifica se uma string de entrada é uma palíndromo.
- k) Elimina: localiza um elemento dentro de uma pilha, desempilhando cada elemento, um a um. Se encontrar, o elemento é removido e a pilha de entrada é restaurada, exceto pelo elemento removido. Se não encontrar, a pilha de entrada é mantida.
- l) Pares_e_impares: empilha uma sequência qualquer de inteiros positivos digitados pelo usuário. Quando o último elemento for inserido, o conteúdo da primeira pilha deve ser distribuído em outras duas pilhas: uma contendo os valores pares e outra contendo os valores ímpares.

2) Implementar a TAD Pilha usando alocação **dinâmica/encadeada**. Operações que a TAD deve contemplar: as mesmas definidas no item anterior, com exceção da operação pilha_cheia.

- 3) Implementar um programa que faça conversões de números inteiros na base 10 para outras bases, de acordo com a opção do usuário. Utilizar a implementação da TAD Pilha usando alocação **estática/sequencial**. Conversões que devem ser previstas:
- a) Decimal para Binário
 - b) [EXTRA] Decimal para Octal
 - c) [EXTRA] Decimal para Hexadecimal.
- 4) Implementar a TAD Fila usando alocação **estática/sequencial e a solução 1 para Lista circular vista em aula** (Ini aponta para posição anterior ao primeiro e despreza uma posição do vetor):
- a) Inicializar_Fila: (vista em sala)
 - b) Fila_é_vazia: (vista em sala)
 - c) Fila_é_cheia: (vista em sala)
 - d) Insere_fila: (vista em sala)
 - e) Remove_fila: (vista em sala)
 - f) Imprimir: imprimir os elementos da fila, do início para o final.
- 5) Implementar a TAD Fila usando alocação **estática/sequencial e uma variação da solução 2 para Lista circular vista em aula** (Utiliza contador de elementos e **apenas** Ini). Operações que a TAD deve contemplar: as mesmas do exercício 4.
- 6) Implementar a TAD Fila usando alocação **dinâmica/encadeada simples (não circular)**. Operações que a TAD deve contemplar: as mesmas do exercício 4.
- 7) Implementar a TAD Fila usando alocação **dinâmica/encadeada circular**. Operações que a TAD deve contemplar: as mesmas do exercício 4.
- 8) [EXTRA] Implementar a TAD Fila de Prioridade Ascendente usando alocação **dinâmica/encadeada e inserção ordenada**. Operações que a TAD deve contemplar:
- a) Inicializar_fpa
 - b) Fpa_é_vazia
 - c) Fpa_é_cheia
 - d) Insere_fpa
 - e) Remove_fpa
 - f) Imprimir: imprimir os elementos da fpa, do início para o final.
- 9) [EXTRA] Implementar a TAD Fila de Prioridade Descendente usando alocação **dinâmica/encadeada e remoção ordenada**. Operações que a TAD deve contemplar, as mesmas do exercício 8 (alterando fpa para fpd)
- 10) [EXTRA] Implementar a TAD Deque usando alocação **estática/sequencial**. Operações que a TAD deve contemplar:
- a) Inicializar_Deque
 - b) Deque_é_vazia
 - c) Deque_é_cheia
 - d) Insere_início_deque
 - e) Insere_final_deque
 - f) Remove_início_deque
 - g) Remove_final_deque
 - h) Imprimir: imprimir os elementos do deque, do início para o final.
- 11) [EXTRA] Implementar a TAD Deque usando alocação **dinâmica/encadeada**. Operações que a TAD deve contemplar: as mesmas do exercício 10.