

UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO  
Avenida Bom Pastor, s/n, Boa Vista - CEP: 55292-270 - Garanhuns/PE

Aluno: \_\_\_\_\_

Disciplina: Algoritmos e Estruturas de Dados I  
Professor: Igor Medeiros Vanderlei  
Curso: Ciência da Computação

Período: II Turno: Noite Semestre: 2022.2

### Pilha (Stack)

Função	Detalhes
push	Insere um elemento no topo da pilha (inserir no fim)
pop	Retira um elemento do topo da pilha (remover do fim)
top	Consulta o valor no topo da pilha, mas não remove.
isEmpty	Retorna 1 caso a pilha esteja vazia e 0 em caso contrário.
print	Exibe todos os valores da pilha. Deve ser utilizada apenas para realização de testes.

1) Considerando as funções na tabela anterior, implemente uma pilha que armazena números inteiros, utilizando a representação sequencial. Faça uma função main para testar o funcionamento da pilha.

2) Considerando as funções na tabela anterior, implemente uma pilha que armazena números inteiros, utilizando a representação encadeada. Faça uma função main para testar o funcionamento da pilha.

3) Escreva um programa que recebe um string do teclado e determina se os parênteses, colchetes e chaves estão bem balanceados. Ex:  $9 * \{ 10 + [2 / 3] + [ (-2) + (6 * 9) ] \} / \{-11\}$

4) Escreva um programa COMPUTACIONALMENTE EFICIENTE que acumula n valores da sequência de Fibonacci em uma pilha. **Observação:** Não utilize função recursiva e evite cálculos repetitivos.

5) Escreva um programa que recebe um string e determina se o string digitado é uma palíndromo. Ex: anilina, ovo, osso, radar, sopapos, socos.

**Obs: Acentos e espaços em branco devem ser desconsiderados.**

Ex: A mãe te ama; luz azul; o galo ama o lago.

6) Resolver a questão “Pilha de Chocolate”:

<https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/chocolate-stack-746c1b56/>

7) Resolver a questão “Biblioteca de Alice”

<https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/algorithm/katrina-and-library-c2ed51f3/>

Estudo complementar – problemas que podem ser resolvidos com pilha:

<https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/>

## Fila (Queue)

Função	Detalhes
enqueue	Adiciona um elemento no fim da fila (enfileirar)
dequeue	Retira um elemento do início da fila (desenfileirar)
peek	Consulta o valor do primeiro elemento da fila, mas não remove.
isEmpty	Retorna 1 caso a fila esteja vazia e 0 em caso contrário.
print	Exibe todos os valores da fila. Deve ser utilizada apenas para realização de testes.

- 1) Considerando as funções na tabela anterior, implemente uma fila que armazena números inteiros, utilizando a representação sequencial. Faça uma função main para testar o funcionamento da fila.
- 2) Considerando as funções na tabela anterior, implemente uma fila que armazena números inteiros, utilizando a representação encadeada. Faça uma função main para testar o funcionamento da fila.
- 3) Escreva uma função que receba duas filas (L1 e L2) e intercala os seus valores, gerando uma terceira fila, L3.

Exemplo:

L1: a1, a2, a3, a4

L2: b1, b2

Saída: L3: a1, b1, a2, b2, a3, a4

- 4) Escreva um programa EFICIENTE que recebe um número inteiro n e imprime os números binários entre 1 e n. Exemplo: para a entrada n=16, a saída deve ser: 1 10 11 100 101 110 111 1000 1001 1010 1011 1100 1101 1110 1111 10000

- 5) Resolver o problema “Torre de Discos”

<https://www.hackerearth.com/practice/data-structures/queues/basics-of-queues/practice-problems/algorithm/disk-tower-b7cc7a50/>

- 6) Resolver o problema de “Esvaziar os Arrays”

<https://www.hackerearth.com/practice/data-structures/queues/basics-of-queues/practice-problems/algorithm/empty-array-31ed638c/>

Estudo complementar:

<https://www.hackerearth.com/practice/data-structures/queues/basics-of-queues/practice-problems/>