



Laboratório 06

1º Escreva um programa que utilize uma pilha para verificar se uma sequência de parênteses está corretamente balanceada. Seu código deve considerar os símbolos: {}, (), [].

Entrada: {[()]}

Entrada: {[()]}

Saída: Entrada Balanceada

Saída: Entrada não Balanceada

2º Crie um programa que use uma pilha para inverter uma string. O usuário deve inserir uma palavra ou frase, e o programa deve exibir o texto invertido.

Entrada: Gatos

Saída: sotaG

3º Implemente um algoritmo que, utilizando as estruturas de pilha, converta expressões matemáticas na notação infixa (exemplo: $3 + 4 * 2$) para a notação pós-fixa (exemplo: $3\ 4\ 2\ * \ +$).

Entrada: $(3 + 4) * 2$

Saída: $3\ 4\ +\ 2\ *$

4º Com base no programa anterior, implemente um programa que solucione, por meio de estrutura de pilha, expressões fornecidas em notação pós-fixa.

Entrada: $3\ 4\ +\ 2\ *$

Saída: 14

5º Crie um programa que simule um sistema de desfazer e refazer utilizando duas pilhas.

- O usuário pode digitar comandos e o programa os armazenará.
- O comando "desfazer" deve remover a última ação realizada.
- O comando "refazer" deve reaplicar a última ação desfeita.

Entrada: "Escrever código"
"Salvar arquivo"
"desfazer"
"refazer"

Saída: Ação desfeita: "Salvar arquivo"
Ação refeita: "Salvar arquivo"

6º Crie um programa que simule o atendimento de clientes em um banco. Utilize uma fila para armazenar os clientes que aguardam atendimento.

- O programa deve permitir adicionar novos clientes à fila.
- Deve ser possível atender um cliente (removendo-o da fila).
- Exibir a fila atual de espera.

Entrada: Adicionar "João"
Adicionar "Victor"
Atender
Adicionar "Caio"

Saída: Atendendo João
Fila atual: ['Victor', 'Caio']

7º Crie um programa que simule uma fila de impressão, onde os documentos são enfileirados para impressão e processados na ordem de chegada.

O programa deve permitir:

- Adicionar novos documentos à fila.
- Processar a impressão (removendo o primeiro documento da fila).
- Exibir os documentos restantes na fila.

Entrada: Adicionar "Documento1.pdf"
Adicionar "Relatório.docx"
Imprimir
Adicionar "Foto.png"

Saída: Imprimindo Documento1.pdf
Fila atual: ['Relatório.docx', 'Foto.png']

8º Implemente um **escalonador de processos** baseado no algoritmo **Round-Robin** utilizando filas.

- Cada processo tem um tempo de execução.
- Cada processo recebe um **quantum de tempo fixo**.
- Se um processo não for concluído dentro do quantum, ele retorna ao final da fila.

Entrada: Adicionar "Processo A" (tempo: 8s)
Adicionar "Processo B" (tempo: 4s)
Adicionar "Processo C" (tempo: 6s)
Quantum: 3s

Saída: Executando Processo A por 3s
Executando Processo B por 3s
Executando Processo C por 3s
Executando Processo A por 3s
Executando Processo C por 3s
Executando Processo A por 2s (finalizado)
Executando Processo B por 1s (finalizado)
Executando Processo C por 3s (finalizado)

9º Implemente uma **fila circular** com tamanho fixo(Array Estático).

- Ao adicionar um novo elemento quando a fila está cheia, o primeiro elemento deve ser sobrescrito.
- Teste sua implementação enfileirando e desenfileirando elementos.

10º Implemente uma fila de prioridade, onde elementos com maior prioridade são atendidos primeiro.

- Cada elemento terá um valor associado à sua prioridade.
- O elemento com a maior prioridade deve ser removido antes dos demais.
- Caso dois elementos tenham a mesma prioridade, o primeiro a chegar é atendido primeiro (FIFO).

Entrada: Adicionar "Paciente A" com prioridade 1

Adicionar "Paciente B" com prioridade 3

Adicionar "Paciente C" com prioridade 2

Saída: Atendendo Paciente B

Atendendo Paciente C

Atendendo Paciente A