

FILA E PILHA EM C

Estrutura de Dados

INTRODUÇÃO

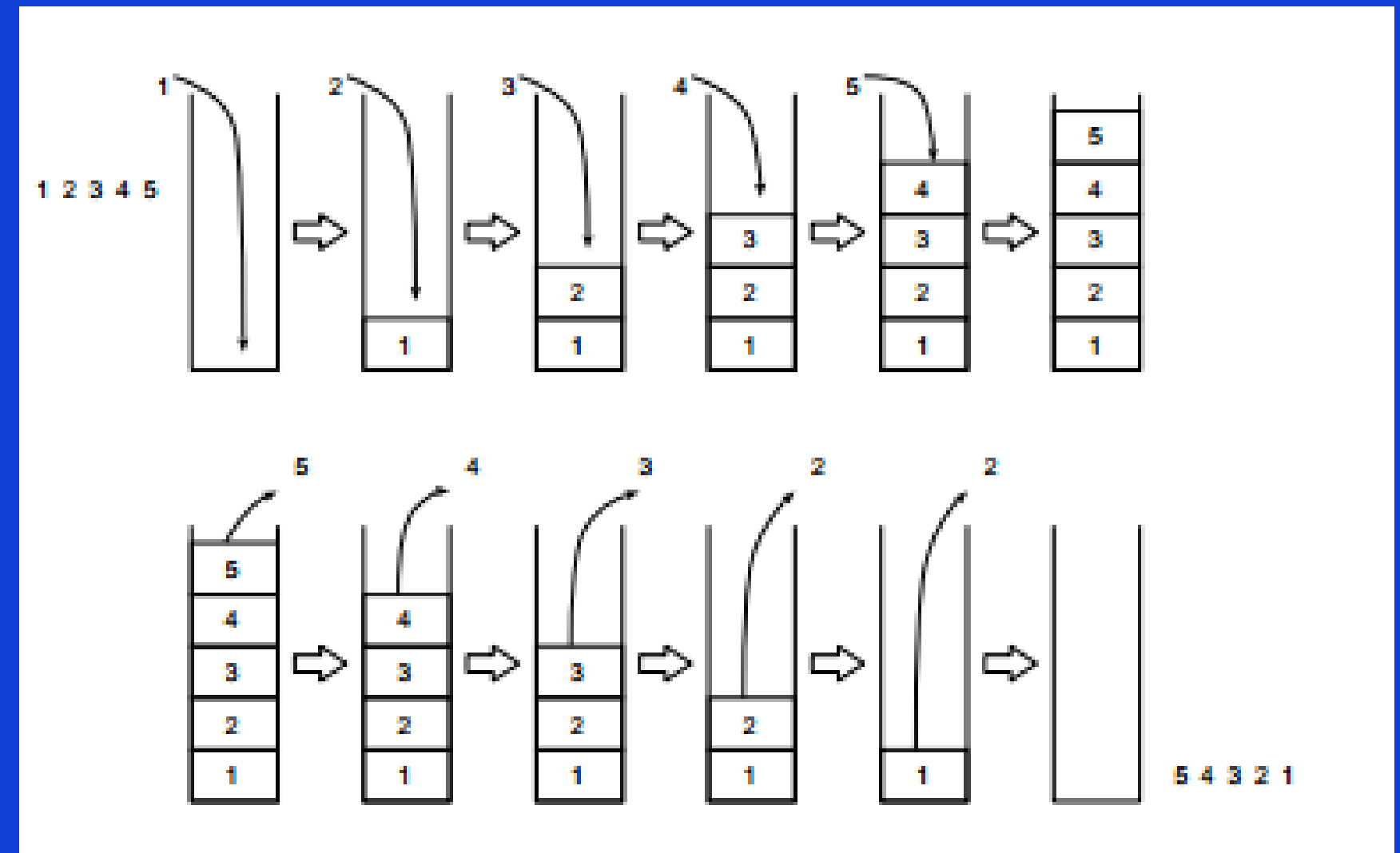
Estruturas de Dados são formas de organizar e armazenar dados de maneira eficiente, permitindo operações específicas com base em requisitos do problema. Duas estruturas de dados essenciais são as pilhas (stacks) e filas (queues). Ambas são amplamente usadas em algoritmos e têm aplicações práticas em sistemas de gerenciamento de dados, programação de tarefas, entre outros.

PILHA (STACK)

A pilha é uma estrutura de dados linear que segue o princípio LIFO (Last In, First Out), ou seja, o último elemento inserido na pilha será o primeiro a ser removido. Imagine uma pilha de pratos: quando você coloca um prato novo, ele vai ao topo, e quando você retira, o último prato colocado é o primeiro a sair.

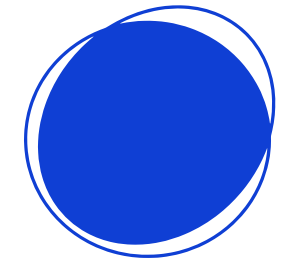
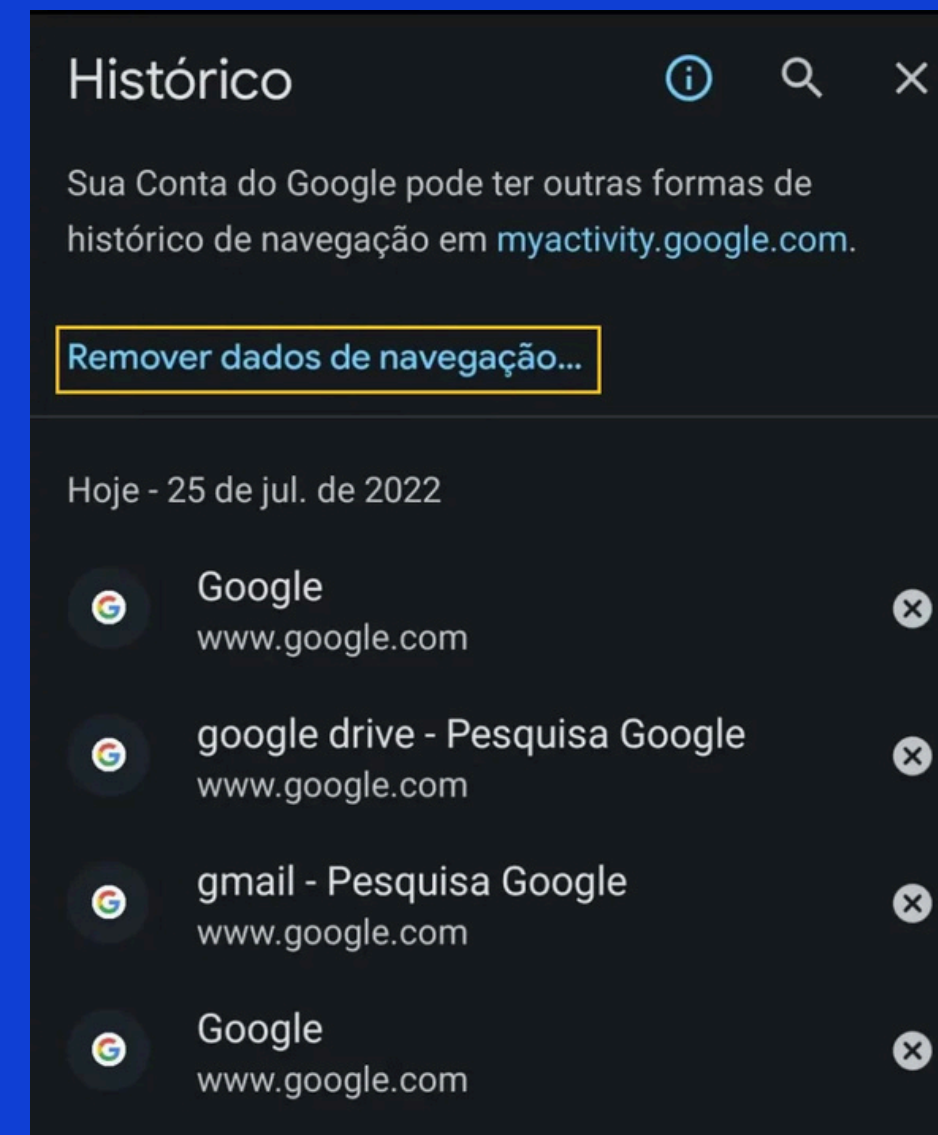
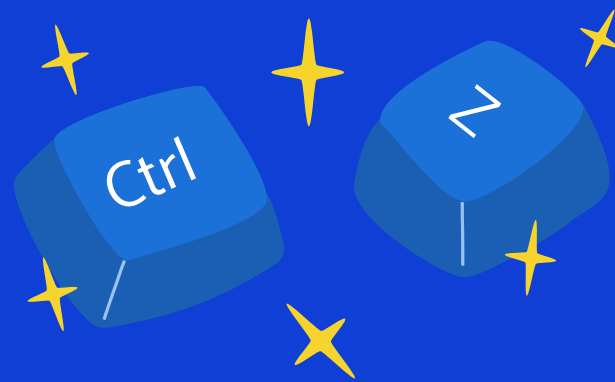
Operações Principais

- **push:** Insere um novo elemento no topo da pilha. Se a pilha estiver cheia (em uma implementação com limite de elementos), ocorre um erro de "pilha cheia".
- **pop:** Remove o elemento do topo da pilha. Se a pilha estiver vazia, ocorre um erro de "pilha vazia".
- **peek ou top:** Acessa o elemento do topo da pilha sem removê-lo.



EXEMPLO DE PILHA NA PRÁTICA

- **Histórico de Navegação:** Em navegadores, o histórico de páginas visitadas funciona como uma pilha. Quando você navega para uma nova página, ela é adicionada ao topo da pilha. Ao pressionar "voltar", o navegador exibe a página do topo, removendo-a da pilha.
- **Desfazer/Refazer:** Ao fazer alterações, cada ação é empilhada. O comando "desfazer" remove a última ação da pilha.



EXEMPLO EM C DE PILHA

```
int main() {
    int opcao, valor;
    do {
        printf("\nEscolha uma opção:\n1. Inserir na pilha\n2. Remover da pilha\n3. Mostrar a pilha\n4. Sair\n");
        scanf("%d", &opcao);
        switch (opcao) {
            case 1:
                printf("Digite um valor para inserir: ");
                scanf("%d", &valor);
                push(&valor);
                break;
            case 2:
                pop();
                break;
            case 3:
                mostrarPilha();
                break;
            case 4:
                printf("Encerrando o programa.\n");
                break;
            default:
                printf("Opção inválida. Tente novamente.\n");
        }
    } while (opcao != 4);
    return 0;
}
```

```
#include <stdio.h>
#define MAX 5

int pilha[MAX], top = -1;

// Função para adicionar um elemento ao topo da pilha
void push(int *valor) {
    if (top == MAX - 1) {
        printf("\nPilha cheia! Não é possível adicionar %d\n", *valor);
    } else {
        pilha[++top] = *valor;
        printf("\n%d adicionado ao topo da pilha.\n", *valor);
    }
}

// Função para remover o elemento do topo da pilha
int pop() {
    if (top == -1) {
        printf("\nPilha vazia! Nenhum elemento para remover.\n");
        return -1;
    } else {
        int removedValue = pilha[top--];
        printf("\n%d removido do topo da pilha.\n", removedValue);
        return removedValue;
    }
}

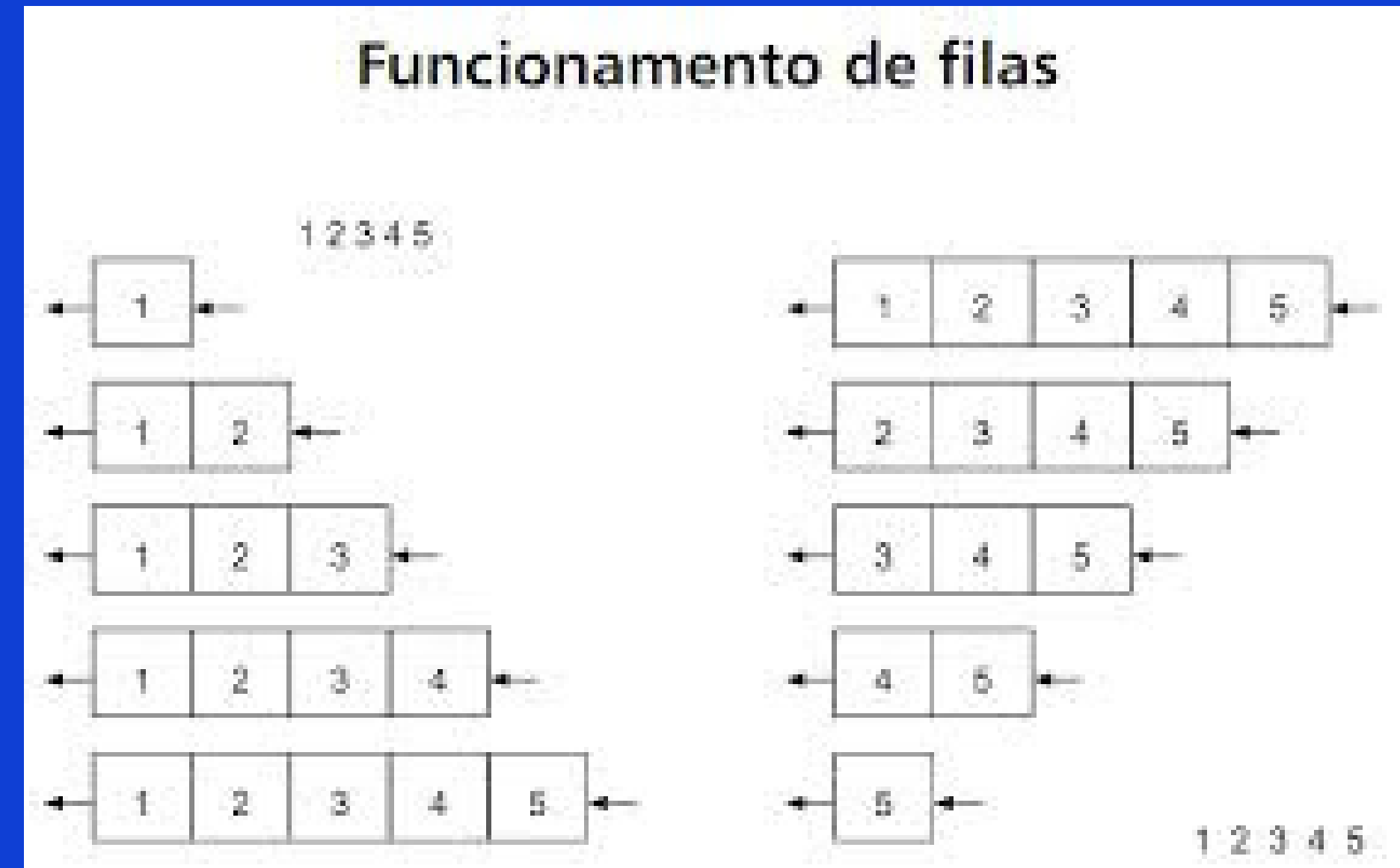
// Função para mostrar todos os elementos da pilha
void mostrarPilha() {
    if (top == -1) {
        printf("\nA pilha está vazia.\n");
    } else {
        printf("\nElementos na pilha: ");
        for (int i = 0; i <= top; i++) {
            printf("%d ", pilha[i]);
        }
        printf("\n");
    }
}
```

FILE (QUEUE)

A fila é uma estrutura de dados linear que segue o princípio FIFO (First In, First Out), onde o primeiro elemento inserido é o primeiro a ser removido, semelhante a uma fila de pessoas: a primeira pessoa a entrar é a primeira a sair.

Operações Principais

- Enqueue: Insere um novo elemento no final da fila. Se a fila estiver cheia, ocorre um erro de "fila cheia".
- Dequeue: Remove o elemento do início da fila. Se a fila estiver vazia, ocorre um erro de "fila vazia".
- Front: Acessa o primeiro elemento da fila sem removê-lo.

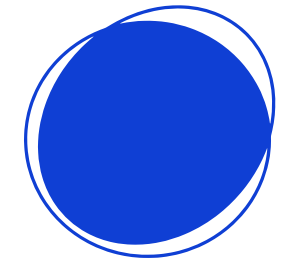


EXEMPLO DE FILA NA PRÁTICA

- **Processamento de Tarefas em Impressoras:** Quando várias tarefas de impressão são enviadas a uma impressora, elas são armazenadas em uma fila de espera. A primeira tarefa enviada é a primeira a ser impressa.



- **Sistemas de Atendimento ao Cliente:** Chamadas telefônicas ou atendimentos seguem a ordem de chegada. A primeira pessoa na fila é atendida primeiro.



EXEMPLO EM C DE FILA

```
int main() {
    int opcao, valor;
    do {
        printf("\nEscolha uma opção:\n1. Inserir na fila\n2. Remover da fila\n3. Mostrar a fila\n4. Sair\n");
        scanf("%d", &opcao);
        switch (opcao) {
            case 1:
                printf("Digite um valor para inserir: ");
                scanf("%d", &valor);
                enqueue(&valor);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                mostrarFila();
                break;
            case 4:
                printf("Encerrando o programa.\n");
                break;
            default:
                printf("Opção inválida. Tente novamente.\n");
        }
    } while (opcao != 4);
    return 0;
}
```

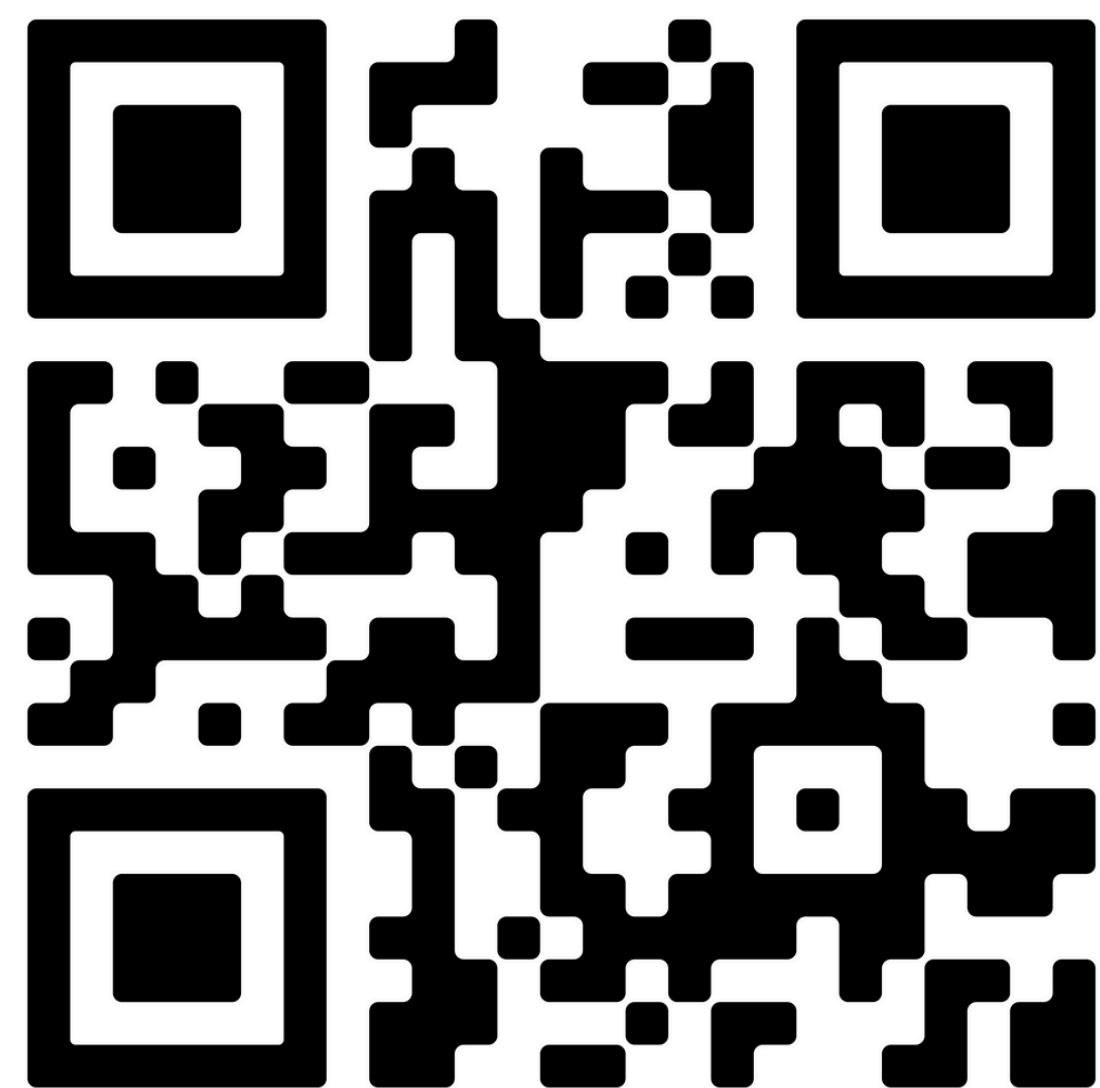
```
#include <stdio.h>
#define MAX 5

int fila[MAX], front = -1, rear = -1;

// Função para adicionar um elemento no final da fila
void enqueue(int *valor) {
    if (rear == MAX - 1) {
        printf("\nFila cheia! Não é possível adicionar %d\n", *valor);
    } else {
        if (front == -1) front = 0;
        fila[++rear] = *valor;
        printf("\n%d adicionado à fila.\n", *valor);
    }
}

// Função para remover o elemento do início da fila
int dequeue() {
    if (front == -1 || front > rear) {
        printf("\nFila vazia! Nenhum elemento para remover.\n");
        return -1;
    } else {
        int removerValor = fila[front++];
        printf("\n%d removido da fila.\n", removerValor);
        return removerValor;
    }
}

// Função para mostrar todos os elementos da fila
void mostrarFila() {
    if (front == -1 || front > rear) {
        printf("\nA fila está vazia.\n");
    } else {
        printf("\nElementos na fila: ");
        for (int i = front; i <= rear; i++) {
            printf("%d ", fila[i]);
        }
        printf("\n");
    }
}
```

OBRIGADO
PELA
ATENÇÃO