

Aluno: Nathalia Coutinho de Souza  
RA: 81401  
Prof: Alan Navia

```
#include <stdio.h>
#include <stdlib.h>

// Definição da estrutura do nó
typedef struct Node {
    int data;
    struct Node* next;
} Node;

// Função para criar um novo nó
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Função para empilhar (push)
void push(Node** top, int data) {
    Node* newNode = createNode(data);
    newNode->next = *top;
    *top = newNode;
    printf("Elemento %d empilhado.\n", data);
}

// Função para desempilhar (pop)
int pop(Node** top) {
    if (*top == NULL) {
        printf("A pilha está vazia.\n");
        return -1;
    }
    Node* temp = *top;
    int data = temp->data;
    *top = temp->next;
    free(temp);
    printf("Elemento %d desempilhado.\n", data);
    return data;
}

// Função para imprimir a pilha
void printStack(Node* top) {
    printf("Pilha: ");
    while (top != NULL) {
```

```

        printf("%d -> ", top->data);
        top = top->next;
    }
    printf("NULL\n");
}

```

```

// Função principal
int main() {
    Node* stack = NULL;

    push(&stack, 10);
    push(&stack, 20);
    push(&stack, 30);

    printStack(stack);

    pop(&stack);
    pop(&stack);

    printStack(stack);

    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>

```

```

// Definição da estrutura do nó
typedef struct Node {
    int data;
    struct Node* next;
} Node;

```

```

// Estrutura da fila
typedef struct Queue {
    Node* front;
    Node* rear;
} Queue;

```

```

// Função para inicializar a fila
Queue* createQueue() {
    Queue* q = (Queue*)malloc(sizeof(Queue));
    q->front = q->rear = NULL;
}

```

```

    return q;
}

// Função para criar um novo nó
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Função para enfileirar (enqueue)
void enqueue(Queue* q, int data) {
    Node* newNode = createNode(data);
    if (q->rear == NULL) {
        q->front = q->rear = newNode;
        printf("Elemento %d enfileirado.\n", data);
        return;
    }
    q->rear->next = newNode;
    q->rear = newNode;
    printf("Elemento %d enfileirado.\n", data);
}

// Função para desenfileirar (dequeue)
int dequeue(Queue* q) {
    if (q->front == NULL) {
        printf("A fila está vazia.\n");
        return -1;
    }
    Node* temp = q->front;
    int data = temp->data;
    q->front = q->front->next;

    if (q->front == NULL) {
        q->rear = NULL;
    }

    free(temp);
    printf("Elemento %d desenfileirado.\n", data);
    return data;
}

// Função para imprimir a fila
void printQueue(Queue* q) {
    Node* temp = q->front;
    printf("Fila: ");
    while (temp != NULL) {

```

```
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

// Função principal
int main() {
    Queue* queue = createQueue();

    enqueue(queue, 10);
    enqueue(queue, 20);
    enqueue(queue, 30);

    printQueue(queue);

    dequeue(queue);
    dequeue(queue);

    printQueue(queue);

    return 0;
}
```