INSTITUTO FEDERAL
Ceará

Nome: Ana Vitória Cabral Duarte
Matrícula: 20212013020205

Avaliação-09

Implementar uma PIlha usando uma lista Encadeada usando as seguintes operações:
1) Verificar o Topo da Pilha. Consiste na verificação da existencia de um elemento na Pilha;
2) Implementar o Push (é a inserção de um elemento na Pilha alterando a informação do Topo);
3) Implementar o Pop (é a retirada de um elemento da Pilha, ou seja é alterado o Topo com o elemnto logo abaixo).
4) Evidenciar a execução das operações acima mostrando o conteúdo da pilha a partir do Topo.

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


// Definição de um nó da pilha
typedef struct Node {
    int data;
    struct Node* next;
} Node;

// Estrutura da pilha
typedef struct Stack {
    Node* top;
} Stack;

// Função para criar uma nova pilha
Stack* createStack() {
    Stack* stack = (Stack*)malloc(sizeof(Stack));
    stack->top = NULL;
    return stack;
}

// Função para verificar se a pilha está vazia
int isEmpty(Stack* stack) {
setlocale(LC_ALL, "portuguese");
```

```
    return stack->top == NULL;
}

// Função para empurrar (push) um elemento para a pilha
void push(Stack* stack, int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->next = stack->top;
    stack->top = newNode;
    printf("Elemento %d empilhado\n", data);
}

// Função para desempilhar (pop) um elemento da pilha
int pop(Stack* stack) {
    if (isEmpty(stack)) {
        printf("A pilha está vazia. Não é possível desempilhar.\n");
        return -1;
    }
    Node* topNode = stack->top;
    int data = topNode->data;
    stack->top = topNode->next;
    free(topNode);
    return data;
}

// Função para mostrar o conteúdo da pilha a partir do topo
void display(Stack* stack) {
    if (isEmpty(stack)) {
        printf("A pilha está vazia.\n");
        return;
    }

    Node* current = stack->top;
    printf("Conteúdo da pilha a partir do topo: ");
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    printf("\n");
}

int main() {
    Stack* stack = createStack();

    display(stack);

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

    display(stack);
```

```
    int topValue = stack->top->data;
    printf("O topo da pilha contém: %d\n", topValue);

    int poppedValue = pop(stack);
    if (poppedValue != -1) {
        printf("Elemento %d desempilhado\n", poppedValue);
    }

    display(stack);

    return 0;
}
```