#include <iostream>

#include <stack>

using namespace std;

void mostrarPilha(stack<int> pilha) {

cout << "Conteúdo da pilha: ";

stack<int> temp = pilha;

while (!temp.empty()) {

cout << temp.top() << " ";

temp.pop();

}

cout << endl;

}

int main() {

stack<int> pilha;

while (true) {

cout << "---------------------------------------\n";

cout << "Menu:\n";

cout << "1. Empilhar\n";

cout << "2. Desempilhar\n";

cout << "3. Ver elemento do topo\n";

cout << "4. Ver pilha completa\n";

cout << "5. Sair\n";

cout << "---------------------------------------\n";

int escolha;

cout << "Escolha uma opção: ";

cin >> escolha;

if (escolha == 1) {

if (pilha.size() < 4) {

int elemento;

cout << "Digite o elemento a empilhar: ";

cin >> elemento;

pilha.push(elemento);

cout << "---------------------------------------\n";

cout << "Elemento " << elemento << " empilhado com sucesso." << endl;

mostrarPilha(pilha);

} else {

cout << "---------------------------------------\n";

cout << "A pilha está cheia. Não é possível empilhar mais elementos." << endl;

}

} else if (escolha == 2) {

if (!pilha.empty()) {

int elementoDesempilhado = pilha.top();

pilha.pop();

cout << "---------------------------------------\n";

cout << "Elemento " << elementoDesempilhado << " desempilhado com sucesso." << endl;

mostrarPilha(pilha);

} else {

cout << "---------------------------------------\n";

cout << "A pilha está vazia. Nada para desempilhar." << endl;

}

} else if (escolha == 3) {

if (!pilha.empty()) {

cout << "Elemento do topo: " << pilha.top() << endl;

} else {

cout << "---------------------------------------\n";

cout << "A pilha está vazia." << endl;

}

} else if (escolha == 4) {

if (!pilha.empty()) {

mostrarPilha(pilha);

} else {

cout << "---------------------------------------\n";

cout << "A pilha está vazia." << endl;

}

} else if (escolha == 5) {

break;

} else {

cout << "---------------------------------------\n";

cout << "Opção inválida. Tente novamente." << endl;

}

}

return 0;

}

---------------------------------------------------------

explicação - roteiro improvisado:

#include <iostream>

#include <stack>

using namespace std;

-> Esta parte do código inclui as bibliotecas iostream e stack, que são necessárias para usar entrada e saída padrão (cin e cout) e implementar uma pilha.

void mostrarPilha(stack<int> pilha) {

cout << "Conteúdo da pilha: ";

stack<int> temp = pilha;

while (!temp.empty()) {

cout << temp.top() << " ";

temp.pop();

}

cout << endl;

}

-> Esta é uma função mostrarPilha que aceita uma pilha como argumento e a exibe na saída padrão. Ela cria uma pilha temporária (temp) para evitar alterar a pilha original. A função percorre a pilha temporária, exibindo os elementos da pilha e, em seguida, imprime uma nova linha para formatação.

int main() {

stack<int> pilha;

-> Aqui, uma pilha de inteiros é criada. Esta pilha será usada para armazenar os elementos.

while (true) {

-> Este é um loop infinito que mantém o programa em execução até que o usuário escolha a opção "Sair".

cout << "Menu:\n";

cout << "1. Empilhar\n";

cout << "2. Desempilhar\n";

cout << "3. Ver elemento do topo\n";

cout << "4. Ver pilha completa\n";

cout << "5. Sair\n";

int escolha;

cout << "Escolha uma opção: ";

cin >> escolha;

-> O código exibe um menu com opções numeradas para o usuário e solicita que o usuário escolha uma opção.

if (escolha == 1) {

-> Se a escolha do usuário for 1, isso significa que ele deseja empilhar um elemento.

if (pilha.size() < 4) {

-> Verifica se a pilha atual tem menos de 4 elementos, para garantir que a pilha não fique cheia.

int elemento;

cout << "Digite o elemento a empilhar: ";

cin >> elemento;

pilha.push(elemento);

cout << "Elemento " << elemento << " empilhado com sucesso." << endl;

mostrarPilha(pilha);

-> O programa solicita que o usuário insira um valor, empilha esse valor na pilha e, em seguida, exibe uma mensagem de sucesso e a pilha atualizada usando a função mostrarPilha.

} else if (escolha == 2) {

-> Se a escolha do usuário for 2, isso significa que ele deseja desempilhar um elemento.

if (!pilha.empty()) {

-> Verifica se a pilha não está vazia, para garantir que haja elementos para desempilhar.

int elementoDesempilhado = pilha.top();

pilha.pop();

cout << "Elemento " << elementoDesempilhado << " desempilhado com sucesso." << endl;

mostrarPilha(pilha);

-> O programa desempilha o elemento do topo da pilha, exibe uma mensagem de sucesso e a pilha atualizada usando a função mostrarPilha.

} else {

cout << "A pilha está vazia. Nada para desempilhar." << endl;

}

-> Se a pilha estiver vazia, o programa informa ao usuário que não há elementos para desempilhar.

} else if (escolha == 3) {

-> Se a escolha do usuário for 3, isso significa que ele deseja ver o elemento do topo da pilha.

if (!pilha.empty()) {

-> Novamente, verifica se a pilha não está vazia.

cout << "Elemento do topo: " << pilha.top() << endl;

-> O programa exibe o elemento do topo da pilha.

} else {

cout << "A pilha está vazia." << endl;

}

-> Se a pilha estiver vazia, o programa informa ao usuário que a pilha está vazia.

} else if (escolha == 4) {

-> Se a escolha do usuário for 4, isso significa que ele deseja ver a pilha completa.

if (!pilha.empty()) {

-> Mais uma vez, verifica se a pilha não está vazia.

mostrarPilha(pilha);

-> O programa chama a função mostrarPilha para exibir todos os elementos da pilha.

} else {

cout << "A pilha está vazia." << endl;

}

-> Se a pilha estiver vazia, o programa informa ao usuário que a pilha está vazia.

} else if (escolha == 5) {

-> Se a escolha do usuário for 5, isso significa que ele deseja sair do programa.

break;

-> O programa sai do loop infinito, encerrando o programa.

} else {

cout << "Opção inválida. Tente novamente." << endl;

}

-> Se o usuário inserir uma opção inválida, o programa informa ao usuário que a opção é inválida e o loop continuará.