**Stack Text Editor**

**CODE:**

#include <iostream> // Input/output ke liye

#include <stack> // Stack data structure ke liye

#include <string> // String operations ke liye

#include <fstream> // File save/load ke liye

using namespace std;

// Global variables

stack<string> undoStack; // Undo ke liye stack

stack<string> redoStack; // Redo ke liye stack

string currentText = ""; // Current text editor ka content

// Function to append new text

void appendText(string newText) {

undoStack.push(currentText); // Pehle current state ko undo stack mein save karo

currentText += newText; // New text ko append karo

while (!redoStack.empty()) { // Redo stack ko clear karo

redoStack.pop();

}

}

// Function to undo last change

void undo() {

if (!undoStack.empty()) {

redoStack.push(currentText); // Current state ko redo stack mein save karo

currentText = undoStack.top(); // Undo stack se previous state lo

undoStack.pop();

} else {

cout << "Undo stack is empty!\n";

}

}

// Function to redo last undone change

void redo() {

if (!redoStack.empty()) {

undoStack.push(currentText); // Current state ko undo stack mein save karo

currentText = redoStack.top(); // Redo stack se next state lo

redoStack.pop();

} else {

cout << "Redo stack is empty!\n";

}

}

// Function to save current text to file

void saveToFile(string filename) {

ofstream file(filename);

if (file.is\_open()) {

file << currentText;

file.close();

cout << "File saved successfully.\n";

} else {

cout << "Unable to open file.\n";

}

}

// Function to load text from file

void loadFromFile(string filename) {

ifstream file(filename);

if (file.is\_open()) {

undoStack.push(currentText); // Current state ko undo stack mein save karo

currentText = "";

string line;

while (getline(file, line)) {

currentText += line + "\n";

}

file.close();

cout << "File loaded successfully.\n";

} else {

cout << "Unable to open file.\n";

}

}

// Main function

int main() {

int choice;

string input;

string filename;

while (true) {

cout << "\n--- Stack Text Editor ---\n";

cout << "1. Append Text\n2. Undo\n3. Redo\n4. Save to File\n5. Load from File\n6. Show Current Text\n0. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

cin.ignore(); // Ignore newline character

switch (choice) {

case 1:

cout << "Enter text to append: ";

getline(cin, input);

appendText(input);

break;

case 2:

undo();

break;

case 3:

redo();

break;

case 4:

cout << "Enter filename to save: ";

getline(cin, filename);

saveToFile(filename);

break;

case 5:

cout << "Enter filename to load: ";

getline(cin, filename);

loadFromFile(filename);

break;

case 6:

cout << "Current Text:\n" << currentText << endl;

break;

case 0:

cout << "Exiting editor...\n";

return 0;

default:

cout << "Invalid choice!\n";

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_