#include <iostream>

#include <fstream>

#include <unordered\_map>

#include <string>

using namespace std;

unordered\_map<string, string> accessCodes;

void displayIntro() {

cout << "\nFile Management Helper\n";

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

cout << "This helper assists with organizing your personal files.\n";

}

void showOptions() {

cout << "\nAvailable actions:\n";

cout << "1. Create new file\n";

cout << "2. Add to existing file\n";

cout << "3. View file contents\n";

cout << "4. Remove file\n";

cout << "5. Exit helper\n";

cout << "Enter action number: ";

}

void handleNewFile() {

string fileName, accessCode;

cout << "\nCreating new file\n";

cout << "Enter file name: ";

getline(cin, fileName);

ifstream existingFile(fileName);

if (existingFile.good()) {

cout << "File already exists with that name.\n";

existingFile.close();

return;

}

cout << "Create access code: ";

getline(cin, accessCode);

ofstream newFile(fileName);

if (newFile.is\_open()) {

newFile.close();

accessCodes[fileName] = accessCode;

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

} else {

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

}

}

bool verifyAccess(const string& fileName) {

if (accessCodes.find(fileName) == accessCodes.end()) {

cout << "File not found or not protected.\n";

return false;

}

for (int tries = 0; tries < 3; tries++) {

string userCode;

cout << "Enter access code (" << 3-tries << " attempts remaining): ";

getline(cin, userCode);

if (userCode == accessCodes[fileName]) {

return true;

}

cout << "Incorrect code.\n";

}

cout << "Access denied.\n";

return false;

}

void appendToFile() {

string fileName, content;

cout << "\nAdd to file\n";

cout << "Enter file name: ";

getline(cin, fileName);

ifstream checkFile(fileName);

if (!checkFile.good()) {

cout << "File does not exist.\n";

return;

}

checkFile.close();

if (verifyAccess(fileName)) {

cout << "Enter content to add:\n";

getline(cin, content);

ofstream file(fileName, ios::app);

if (file.is\_open()) {

file << content << "\n";

file.close();

cout << "Content added.\n";

} else {

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

}

}

}

void displayFile() {

string fileName;

cout << "\nView file\n";

cout << "Enter file name: ";

getline(cin, fileName);

ifstream file(fileName);

if (!file.good()) {

cout << "File does not exist.\n";

return;

}

file.close();

if (verifyAccess(fileName)) {

ifstream readFile(fileName);

string line;

cout << "\nFile contents:\n";

while (getline(readFile, line)) {

cout << line << endl;

}

readFile.close();

}

}

void deleteFile() {

string fileName;

cout << "\nRemove file\n";

cout << "Warning: This cannot be undone.\n";

cout << "Enter file name: ";

getline(cin, fileName);

ifstream file(fileName);

if (!file.good()) {

cout << "File does not exist.\n";

return;

}

file.close();

if (verifyAccess(fileName)) {

cout << "Confirm removal (yes/no): ";

string confirm;

getline(cin, confirm);

if (confirm == "yes") {

if (remove(fileName.c\_str()) == 0) {

accessCodes.erase(fileName);

cout << "File removed.\n";

} else {

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

}

} else {

cout << "Action cancelled.\n";

}

}

}

void runHelper() {

displayIntro();

while (true) {

showOptions();

int choice;

cin >> choice;

cin.ignore();

switch (choice) {

case 1:

handleNewFile();

break;

case 2:

appendToFile();

break;

case 3:

displayFile();

break;

case 4:

deleteFile();

break;

case 5:

cout << "\nHelper session ended.\n";

return;

default:

cout << "Invalid selection.\n";

}

}

}

int main() {

runHelper();

return 0;

}