Lab 13 – File Handling

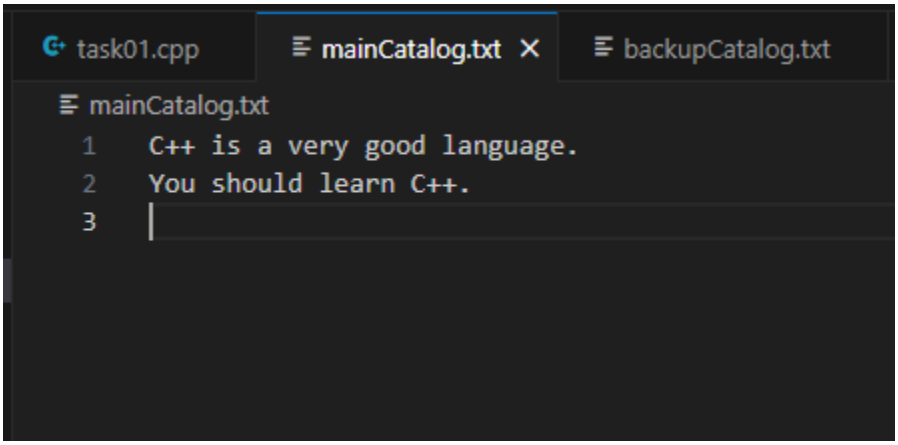
Task 01:

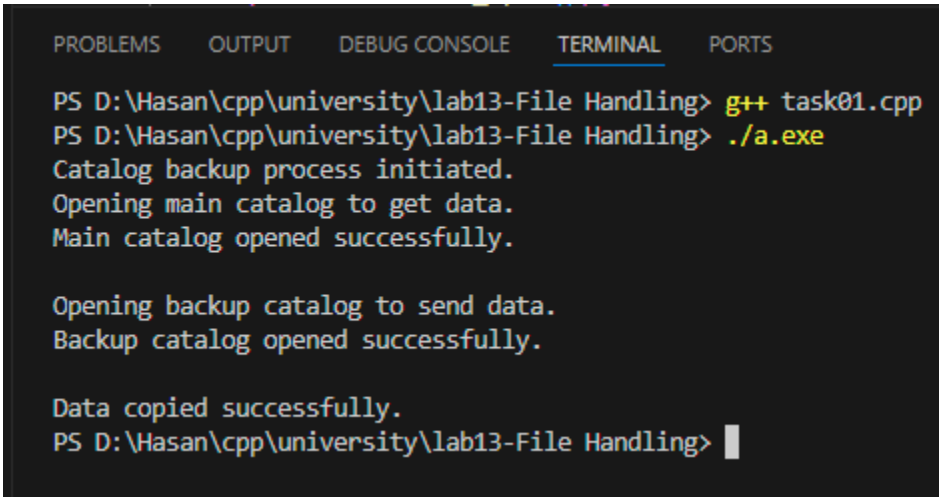
You are tasked with creating a program to back up the library's catalog. The program should copy the contents of the main catalog file to a backup file, ensuring that the library's data is securely preserved in case of any unforeseen events.

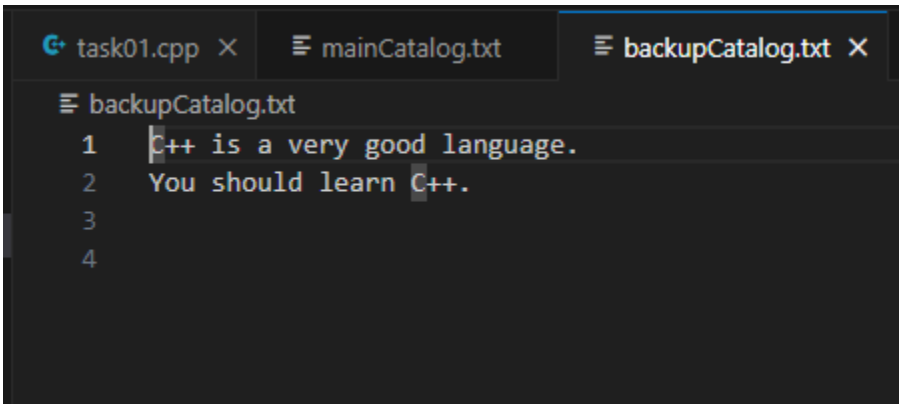
Code:

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;

int main(){
    string row;
    cout << "Catalog backup process initiated." << endl;

    cout << "Opening main catalog to get data." << endl;
    ifstream mainFileIn("mainCatalog.txt", ios::in);
    if(!mainFileIn.is_open()){
        cout << "Error in opening main catalog." << endl;
    }
    else{
        cout << "Main catalog opened successfully." << endl;
    }

    cout << "\nOpening backup catalog to send data." << endl;
    ofstream backupFileOut("backupCatalog.txt", ios::app);
    if(!backupFileOut) cout << "Error in opening backup catalog." << endl;
    else{
        cout << "Backup catalog opened successfully." << endl;
    }

    while(!mainFileIn.eof()){
        getline(mainFileIn, row);
        backupFileOut << row << endl;
    }
    mainFileIn.close();
    backupFileOut.close();
    cout << "\nData copied successfully." << endl;
}
```

Output:


task01.cpp
mainCatalog.txt
backupCatalog.txt
mainCatalog.txt
1 C++ is a very good language.
2 You should learn C++.
3


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS D:\Hasan\cpp\university\lab13-File Handling> g++ task01.cpp
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
Catalog backup process initiated.
Opening main catalog to get data.
Main catalog opened successfully.
Opening backup catalog to send data.
Backup catalog opened successfully.
Data copied successfully.
PS D:\Hasan\cpp\university\lab13-File Handling>


task01.cpp
mainCatalog.txt
backupCatalog.txt
backupCatalog.txt
1 C++ is a very good language.
2 You should learn C++.
3
4

Task 02:

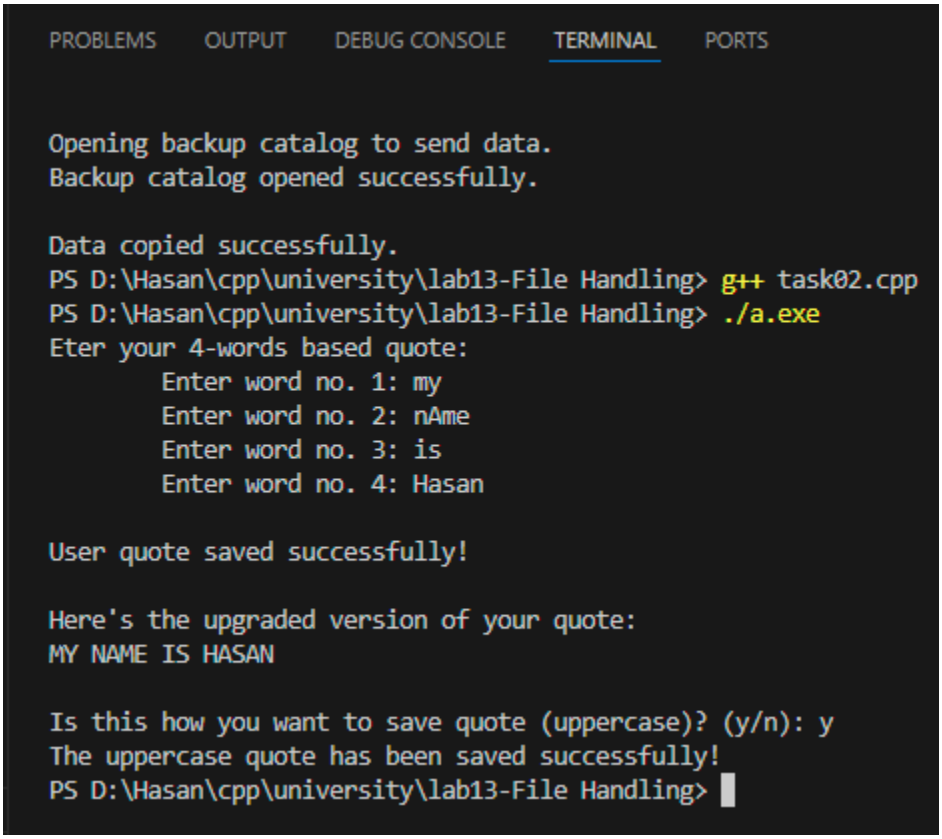
You're developing a motivational quote generator where users can input a four-word quote, which will be permanently stored. Since users may prefer mixed-case quotes, you'll present the final output for review before saving it. After storing the words, the program will display the quote reconstructed with all words in uppercase, providing a visual representation of how the quote would look with case conversion.

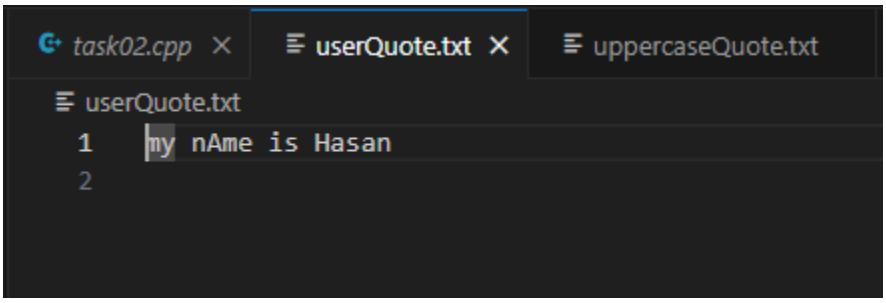
Code:

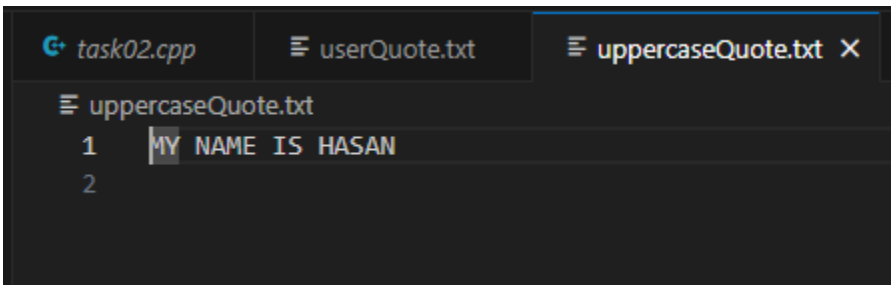
```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;

int main(){
    string word[4];
    char choice;

    // getting quote fom user word by word
    cout << "Eter your 4-words based quote:" << endl;
    for(int i = 0; i < 4; i++){
        cout << "\tEnter word no. "<<(i+1)<<": ";
        getline(cin, word[i]);
    }

    //saving user's entered quote into file
    ofstream userOut("userQuote.txt", ios::app);
    if(!userOut.is_open()) cout << "Error in opening file to save user quote." <<
endl;
    else{
        for(int i = 0; i < 4; i++){
            userOut << word[i] << " ";
        }
        userOut << endl;
        cout << "\nUser quote saved successfully!" << endl;
        userOut.close();
    }
    // converting the quote unto uppercase
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < word[i].length(); j++){
            if(word[i][j] >= 97 && word[i][j] <= 122){
                word[i][j] -= 32;
            }
        }
    }
```

```
    // displaying the uppercase quote, & asking for verification
    cout << "\nHere's the upgraded version of your quote: " << endl;
    for(int i = 0; i < 4; i++){
        cout << word[i] << " ";
    }
    cout << endl;
    do{
        cout << "\nIs this how you want to save quote (uppercase)? (y/n): ";
        cin >> choice;

        if(choice != 'y' && choice != 'n'){
            cout << "\nPlease eneter valid choice! (y/n)" << endl;
        }

    }while(choice != 'y' && choice != 'n');

    // if approved by user, saving it into file
    if(choice == 'y'){
        ofstream upperOut("uppercaseQuote.txt", ios::app);
        if(!upperOut.is_open()) cout << "Error in opening file to save uppercase
quote!" << endl;
        else{
            for(int i = 0; i < 4; i++){
                upperOut << word[i] << " ";
            }
            upperOut << endl;
            cout << "The uppercase quote has been saved successfully!" << endl;
            upperOut.close();
        }
    }
    else{
        cout << "\nTerminating program without saving..." << endl;
    }

}
```

Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Opening backup catalog to send data.
Backup catalog opened successfully.
Data copied successfully.
PS D:\Hasan\cpp\university\lab13-File Handling> g++ task02.cpp
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
Eter your 4-words based quote:
Enter word no. 1: my
Enter word no. 2: nAme
Enter word no. 3: is
Enter word no. 4: Hasan
User quote saved successfully!
Here's the upgraded version of your quote:
MY NAME IS HASAN
Is this how you want to save quote (uppercase)? (y/n): y
The uppercase quote has been saved successfully!
PS D:\Hasan\cpp\university\lab13-File Handling>


task02.cpp
userQuote.txt
uppercaseQuote.txt
userQuote.txt
1 my nAme is Hasan
2


task02.cpp
userQuote.txt
uppercaseQuote.txt
uppercaseQuote.txt
1 MY NAME IS HASAN
2

Task 04:

Imagine you're a teacher managing student grades for your class. You need a program to update the grades of your students stored in a text file. The program will prompt you to enter a number representing the amount to increase the grades by. It will then read the student data from the file, increment each student's grade by the specified amount, and display the updated grades.

Code:

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;

struct student{
    string name;
    int marks = 0;
};

void addData(){
    student s;
    cout << endl;
    cout << "Enter student's name: ";
    getline(cin, s.name);
    cout << "Enter marks: ";
    cin >> s.marks;
    cin.ignore();

    ofstream fileOut("studentData.txt", ios::app);
    if(!fileOut.is_open()) cout << "Error in opening file to send student data."
<< endl;
    else{
        fileOut << s.name << "|" << s.marks << endl;
        fileOut.close();
        cout << "Student recxorded siuccessfully!" << endl;
    }
}

void updateData(){
    student oldData;
    int increment = 0;
    string row;
    cout << endl;
    cout << "Enter marks which you want to add: ";
    cin >> increment;
```

```
    ifstream stuIn("studentData.txt", ios::in);
    ofstream tempOut("temp.txt", ios::app);

    if(!stuIn.is_open()) cout << "Error in opening file to read student data." <<
endl;
    else{
        while(!stuIn.eof()){
            getline(stuIn, row);
            if(row.empty()) continue;

            int pos1 = row.find('|');

            oldData.name = row.substr(0, pos1);
            oldData.marks = stoi(row.substr(pos1 + 1));

            tempOut << oldData.name << "|" << (oldData.marks + increment) <<
endl;
        }

        tempOut.close();
        stuIn.close();

        remove("studentData.txt");
        rename("temp.txt", "studentData.txt");
    }
}

int main(){
    int choice = 0;
    cout << "***Welcome to Grades Updating App***" << endl;
    do{
        cout << "\n\t---MENU---" << endl;
        cout << "1. Add data." << endl;
        cout << "2. Update data." << endl;
        cout << "3. Exit." << endl;
        cout << "Enter choice: ";
        cin >> choice;
        cin.ignore();

        switch(choice){
            case 1:
            addData();
            break;
            case 2:
            updateData();
```

```
            break;
            case 3:
            cout << "\nClosing program..." << endl;
            break;
            default:
            cout << "\nInvalid choice!" << endl;
            break;
        }
    }while(choice != 3);
}
```

Output:

```
PS D:\Hasan\cpp\university\lab13-File Handling> ./a.exe
***Welcome to Grades Updating App***

        ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 1

Enter student's name: Muhammad Hasan
Enter marks: 10
Student recxorded siuccessfully!

        ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 2

Enter marks which you want to add: 5

        ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 1

Enter student's name: Random
Enter marks: 20
Student recxorded siuccessfully!

        ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 2
```

```
Enter choice: 2

Enter marks which you want to add: 5

        ---MENU---
1. Add data.
2. Update data.
3. Exit.
Enter choice: 3

Closing program...
PS D:\Hasan\cpp\university\lab13-File Handling>
```

task04.cpp studentData.txt

studentData.txt

```
1   Muhammad Hasan|20
2   Random|25
3
```