Part 1:

*/\**

*Description: This program reads data from a file, encrypts it, writes the encrypted data to another file, and can return the decrypted data*

*Author: Omer Ahmer*

*COMSC 165-5065*

*Date: 10/25/22*

*Status: Complete*

*\*/*

**#include** <iostream>

**#include** <fstream>

**#include** <cstdlib>

**#include** <string>

**using** **namespace** std;

*// function to get data from file and encrypt*

**void** encrypt(string filename);

*// function to decrypt from other file*

**void** decrypt(string source);

**int** main() {

string source;

cout **<<** "Enter source filename: ";

getline(cin, source);

encrypt(source);

}

**void** encrypt(string filename) {

string toEncrypt;

ifstream input;

input.open(filename);

**if** (input) {

*// attempt to read first line of file*

getline(input, toEncrypt);

**while** (input) { *// while we have successfully read a line of text*

cout **<<** "The original string is: " **<<** toEncrypt **<<** endl;

getline(input, toEncrypt); *// attempt to get the next line*

}

input.close();

}

**else** {

cout **<<** "Could not open file" **<<** endl;

}

**for** (**int** i **=** 0; i **<** toEncrypt.size(); i**++**) {

toEncrypt**[**i**]** **+=** 10;

}

ofstream out("encrypted.txt");

out **<<** toEncrypt;

out.close();

decrypt("encrypted.txt");

}

**void** decrypt(string source) {

ifstream input;

string toDecrypt;

input.open(source);

**if** (input) {

getline(input, toDecrypt);

**while** (input) {

cout **<<** "The encrypted string is: " **<<** toDecrypt **<<** endl;

getline(input, toDecrypt);

}

input.close();

}

**else** {

cout **<<** "Could not open file" **<<** endl;

}

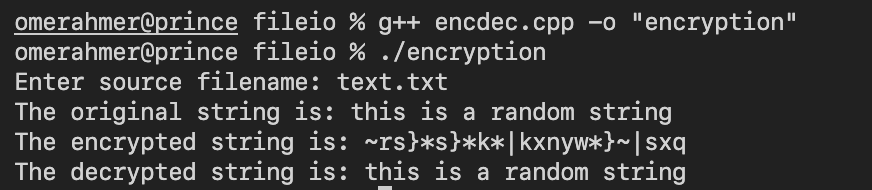
**for** (**int** i **=** 0; i **<** toDecrypt.size(); i**++**) {

toDecrypt**[**i**]** **-=** 10;

}

cout **<<** "The decrypted string is: " **<<** toDecrypt **<<** endl;

}



Part 2:

*/\**

*Description: This program creates a structure of inventory items and can input and return data from the inventory*

*Author: Omer Ahmer*

*COMSC 165-5065*

*Date: 10/26/22*

*Status: Complete*

*\*/*

**#include** <iostream>

**#include** <fstream>

**#include** <string>

**#include** <sstream>

**using** **namespace** std;

**struct** Inventory {

string name;

**int** quantity;

**double** w\_price, r\_price;

};

**bool** getItem(Inventory **&**item);

**void** addRecord(string filename);

**void** displayRecord(fstream **&**file, Inventory **&**item);

**void** changeRecord(fstream **&**file, Inventory **&**item);

**void** displayAll(fstream **&**file);

**void** Report(fstream **&**file);

**int** main() {

**int** input;

string filename **=** "inventory.txt";

fstream File("inventory.txt", ios::in **|** ios::out **|** ios::binary);

Inventory Item;

cout **<<** "1. Add new records to the file" **<<** endl;

cout **<<** "2. Display a record in the file" **<<** endl;

cout **<<** "3. Change a record in the file" **<<** endl;

cout **<<** "4. Display all records" **<<** endl;

cout **<<** "5. Display report" **<<** endl;

cout **<<** "6. Exit" **<<** endl;

cin **>>** input;

**switch** (input) {

**case** 1:

addRecord(filename);

**break**;

**case** 2:

displayRecord(File, Item);

**break**;

**case** 3:

changeRecord(File, Item);

**break**;

**case** 4:

displayAll(File);

**break**;

**case** 5:

Report(File);

**break**;

**case** 6:

**break**;

}

}

**bool** getItem(Inventory **&**item) {

string name;

**int** quantity;

**double** w\_cost, r\_cost;

cout **<<** "Enter item name:" **<<** endl;

getline(cin, name);

cout **<<** "Enter quantity of item:" **<<** endl;

cin **>>** quantity;

**if** (quantity **<** 0) {

cout **<<** "You have entered an invalid quantity." **<<** endl;

**return** **false**;

}

cout **<<** "Enter wholesale cost:" **<<** endl;

cin **>>** w\_cost;

**if** (w\_cost **<** 0) {

cout **<<** "You have entered an invalid wholesale price." **<<** endl;

**return** **false**;

}

cout **<<** "Enter retail cost:" **<<** endl;

cin **>>** r\_cost;

**if** (r\_cost **<** 0) {

cout **<<** "You have entered an invalid retail price." **<<** endl;

**return** **false**;

}

item.name **=** name;

item.quantity **=** quantity;

item.w\_price **=** w\_cost;

item.r\_price **=** r\_cost;

**return** **true**;

}

**void** addRecord(string filename) {

Inventory item;

**if** (getItem(item)) {

fstream file;

file.open("inventory.txt", ios::in**|**ios::out**|**ios::binary);

**if** (file) {

file **<<** item.name **<<** ", " **<<** item.quantity **<<** ", " **<<** item.w\_price **<<** ", " **<<** item.r\_price;

file.close();

}

**else** {

cout **<<** "Unable to open file." **<<** endl;

}

}

**else** {

cout **<<** "Unable to add new record." **<<** endl;

}

}

**void** displayRecord(fstream **&**file, Inventory **&**item) {

**int** recordNum;

cout **<<** "Enter record number:" **<<** endl;

cin **>>** recordNum;

file.read(**reinterpret\_cast<char\*>**(**&**item), **sizeof**(item));

cout **<<** "Record number: " **<<** recordNum **<<** endl;

cout **<<** "Item name: " **<<** item.name **<<** endl;

cout **<<** "Quantity: " **<<** item.quantity **<<** endl;

cout **<<** "Wholesale price: " **<<** item.w\_price **<<** endl;

cout **<<** "Retail price: " **<<** item.r\_price **<<** endl;

}

**void** changeRecord(fstream **&**file, Inventory **&**item) {

file.write(**reinterpret\_cast<char\*>**(**&**item), **sizeof**(item));

cout **<<** "Enter item name: " **<<** endl;

getline(cin, item.name);

cout **<<** "Enter item quantity:";

cin **>>** item.quantity;

cout **<<** "Enter wholesale price:";

cin **>>** item.w\_price;

cout **<<** "Enter retail price:";

cin **>>** item.r\_price;

}

**void** displayAll(fstream **&**file) {

string strings;

**int** count **=** 1;

**while** (getline(file, strings)) {

cout **<<** "Contents of file " **<<** count **<<** ":" **<<** endl **<<** strings **<<** endl;

count**++**;

}

cout **<<** "The total number of records is " **<<** count **<<** endl;

}

**void** Report(fstream **&**file) {

string strings[4];

**int** totalQ **=** 0, totalW **=** 0, totalR **=** 0;

**while** (getline(file, strings)) {

**int** q, w, r;

stringstream info(strings[0] **+** " " **+** strings[1] **+** " " **+** strings[3]);

info **>>** q **>>** w **>>** r;

totalQ **+=** q;

totalW **+=** w;

totalR **+=** r;

}

cout **<<** "Total quantity of items: " **<<** totalQ **<<** endl;

cout **<<** "Total wholesale price of items: " **<<** totalW **<<** endl;

cout **<<** "Total retail price of items: " **<<** totalR **<<** endl;

}