```cpp
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <thread>
#include <chrono>

using namespace std;

// Define a class to represent book information
class Book {
public:
    // Book title
    string title;
    // Book author
    string author;
    // Book genre
    string genre;
    // Book publication year
    int year;

    // Constructor to initialize book attributes
    Book(const string& _title, const string& _author, const string& _genre, int _year)
        : title(_title), author(_author), genre(_genre), year(_year) {}
};

// Function to perform insertion sort on a vector of books
void insertionSort(vector<Book>& books) {
    // Get the number of books in the vector
    int n = books.size();
    // Iterate over the vector, starting at the second book
    for (int i = 1; i < n; i++) {
        // Store the current book in a temporary variable
        Book temp = books[i];
        // Initialize the index of the previous book
        int j = i - 1;

        // Compare books based on their publication year and swap if necessary
        while (j >= 0 && books[j].year > temp.year) {
            books[j + 1] = books[j];
            j--;
        }

        // Place the current book in its sorted position
        books[j + 1] = temp;
    }
}

void playSiren() {
    for (int i = 0; i < 10; i++) {
        cout << "\a";  // Produces a beep sound (system-dependent)
        this_thread::sleep_for(chrono::milliseconds(500));  // Sleep for 500 milliseconds
    }
}

int main() {
    int workIDs[10] = {44306237, 44306232, 44306238, 44306239, 44306236, 44306235, 44306234, 44306233, 44306232, 44306231};
    int passwords[10] = {12345, 12345, 22345, 32345, 42345, 52345, 62345, 72345, 82345, 92345};
    int attempt = 3;
    bool isAuthenticated = false;

    while (attempt > 0) {
        int userWorkID;
        int userPassword;

        cout << "Please Enter Your Work ID: " << endl;
        cin >> userWorkID;
        cout << "Please Enter Your Password: " << endl;
        cin >> userPassword;

        isAuthenticated = false;

        for (int i = 0; i < 10; i++) {
            if (workIDs[i] == userWorkID && passwords[i] == userPassword) {
                isAuthenticated = true;
                break;
            }
        }

        if (isAuthenticated) {
            cout << "ACCESS GRANTED." << endl;
            break;
        } else {
            cout << "\a"; // Beep sound for incorrect password
            cout << "ACCESS DENIED!!! Please Try again" << endl;
            attempt--;

            if (attempt == 0) {
                cout << "Attempts Run Out. Access Locked!" << endl;
                // Play the siren sound if access is locked
                playSiren();
                return 0;  // Exit the program
            } else {
                cout << attempt << " attempts Remaining" << endl;
            }
        }
    }

    // Create a vector to store the books
    vector<Book> bookList;

    cout << "Welcome to the Yellow Pages for Books!" << endl;

    while (true) {
        cout << "\nEnter book details or type 'q' to quit:" << endl;
        string title, author, genre;
        int year;

        cout << "Title: ";
        cin.ignore();
        getline(cin, title);

        if (title == "q") {
            break;
        }

        cout << "Author: ";
        getline(cin, author);

        cout << "Genre: ";
        getline(cin, genre);

        cout << "Year of Publication: ";
        cin >> year;

        bookList.push_back(Book(title, author, genre, year));
        insertionSort(bookList);
        cout << "Book added successfully!" << endl;
```

```cpp
    }

    cout << "\nBooks sorted by publication year:" << endl;

    for (const Book& book : bookList) {
        cout << "Title: " << book.title << ", Year: " << book.year << endl;
    }


    return 0;
}


```