```
#include<iostream>
#include<cstring>
using namespace std;

int n;
int f;
class book {
    int cost;
    char author[10], title[10];
public:
    void accept();
    void display();
    void ascend();
    void descend();
    void less500();
    void more500();
    void deleteT();
    void deleteWT();
} B[100];

void book::accept() {
    cout << "Enter the title: ";
    cin >> title;
    cout << "Enter the author: ";
    cin >> author;
    cout << "Enter the cost: ";
    cin >> cost;
}

void book::display() {

    cout << "\n" << title << "\t" << author << "\t" << cost;
}

void book::ascend() {
    book T;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (B[i].cost > B[j].cost) {
                T = B[i];
                B[i] = B[j];
                B[j] = T;
            }
        }
    }
    cout << "\nBooks sorted in ascending order of cost:";
    for (int i = 0; i < n; i++) {
        B[i].display();
```

```
    }
}

void book::descend() {
    book T;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (B[i].cost < B[j].cost) {
                T = B[i];
                B[i] = B[j];
                B[j] = T;
            }
        }
    }
    cout << "\nBooks sorted in descending order of cost:";
    for (int i = 0; i < n; i++) {
        B[i].display();
    }
}

void book::less500() {
    cout << "\nBooks costing less than 500:";
    for (int i = 0; i < n; i++) {
        if (B[i].cost < 500) {
            B[i].display();
        }
    }
}

void book::more500() {
    int count = 0;
    cout << "\nBooks costing more than 500:";
    for (int i = 0; i < n; i++) {
        if (B[i].cost > 500) {
            B[i].display();
            count++;
        }
    }
    cout << "\nTotal number of books having cost more than 500: " << count;
}

void book::deleteT() {
    book C[100];
    int j = 0;
    for (int i = 0; i < n; i++) {
       f=0;
        for (int k = 0; k < j; k++) {
            if (strcmp(B[i].title, C[k].title) == 0 && strcmp(B[i].author, C[k].author) == 0) {
```

```
            f=1;
               break;
            }
         }
         if (!f==1) {
            C[j++] = B[i];
         }
      }
      cout << "\nBooks after deleting duplicates based on title and author:";
      for (int i = 0; i < j; i++) {
         C[i].display();
      }
      n = j;
}

void book::deleteWT() {
      for (int i = 0; i < n; i++) {
         for (int j = i + 1; j < n;) {
            if (strcmp(B[i].title, B[j].title) == 0 && strcmp(B[i].author, B[j].author) == 0) {

               for (int k = j; k < n - 1; k++) {
                  B[k] = B[k + 1];
               }
               n--;
            } else {
               j++;
            }
         }
      }
      cout << "\nBooks after deleting duplicates based on title and author (without using
temporary array):";
      for (int i = 0; i < n; i++) {
         B[i].display();
      }
}

int main() {
      int ch;
      cout << "Enter total number of entries: ";
      cin >> n;

      for (int i = 0; i < n; i++) {
         B[i].accept();
      }

      do {
         cout << "\n\nMenu:";
         cout << "\n1. Display all books";
```

```
        cout << "\n2. Sort books in ascending order of cost";
        cout << "\n3. Sort books in descending order of cost";
        cout << "\n4. Display books costing less than 500";
        cout << "\n5. Display books costing more than 500";
        cout << "\n6. Delete duplicates using a temporary array";
        cout << "\n7. Delete duplicates without using a temporary array";
        cout << "\n8. Exit";
        cout << "\nEnter your choice: ";
        cin >> ch;

        switch (ch) {
            case 1:
                cout << "\n" << "Title" << "\t" << "Author" << "\t" << "Cost";
                for (int i = 0; i < n; i++) {
                    B[i].display();
                }
                break;
            case 2:
                B[0].ascend();
                break;
            case 3:
                B[0].descend();
                break;
            case 4:
                B[0].less500();
                break;
            case 5:
                B[0].more500();
                break;
            case 6:
                B[0].deleteT();
                break;
            case 7:
                B[0].deleteWT();
                break;
            case 8:
                cout << "Exiting program.\n";
                break;
            default:
                cout << "Invalid choice, please try again.\n";
        }
    } while (ch != 8);

    return 0;
}
```