Write C++ program using STL for sorting and searching user defined records such as Item records (Item code, name, cost, quantity etc) using vector container

Program-

```
#include <iostream>
#include <vector>
#include <algorithm>  // For sort and find
#include <string>
using namespace std;

// Define the Item class to represent an item record
class Item {
public:
   int itemCode;
   string itemName;
   double itemCost;
   int itemQuantity;

   // Constructor to initialize Item data
   Item(int code, string name, double cost, int quantity)
      : itemCode(code), itemName(name), itemCost(cost), itemQuantity(quantity) {}

   // Function to display the item details
   void display() const {
      cout << "Item Code: " << itemCode << ", "
         << "Item Name: " << itemName << ", "
         << "Cost: $" << itemCost << ", "
         << "Quantity: " << itemQuantity << endl;
   }
};

// Function to compare two items based on item code
bool compareByCode(const Item& a, const Item& b) {
   return a.itemCode < b.itemCode;
}

// Function to compare two items based on cost
bool compareByCost(const Item& a, const Item& b) {
   return a.itemCost < b.itemCost;
}

// Main function
int main() {
   // Create a vector to store Item records
   vector<Item> inventory;

   // Adding some items to the inventory
   inventory.push_back(Item(101, "Apple", 0.5, 100));
   inventory.push_back(Item(102, "Banana", 0.3, 150));
   inventory.push_back(Item(103, "Orange", 0.7, 80));
```

```
inventory.push_back(Item(104, "Mango", 1.5, 50));
inventory.push_back(Item(105, "Grapes", 2.0, 30));

// Displaying unsorted inventory
cout << "Unsorted Inventory:" << endl;
for (const auto& item : inventory) {
    item.display();
}

// Sorting the inventory by item code
sort(inventory.begin(), inventory.end(), compareByCode);

// Displaying sorted inventory by item code
cout << "\nInventory Sorted by Item Code:" << endl;
for (const auto& item : inventory) {
    item.display();
}

// Sorting the inventory by cost
sort(inventory.begin(), inventory.end(), compareByCost);

// Displaying sorted inventory by cost
cout << "\nInventory Sorted by Item Cost:" << endl;
for (const auto& item : inventory) {
    item.display();
}

// Searching for an item by item code using std::find
int searchCode = 103;
auto it = find_if(inventory.begin(), inventory.end(), [searchCode](const Item& item) {
    return item.itemCode == searchCode;
});

if (it != inventory.end()) {
    cout << "\nItem Found: ";
    it->display();
} else {
    cout << "\nItem with code " << searchCode << " not found!" << endl;
}

// Searching for an item by name
string searchName = "Mango";
it = find_if(inventory.begin(), inventory.end(), [searchName](const Item& item) {
    return item.itemName == searchName;
});

if (it != inventory.end()) {
    cout << "\nItem Found: ";
    it->display();
} else {
    cout << "\nItem with name " << searchName << " not found!" << endl;
```

```
    }

    return 0;
}
```