In stock market, the user buys/sells stocks, and each buy/sell action is called an `order`. An order includes an user id, the stock symbol the user wants to trade, the trading amount (how many shares of stocks) with what price. For example an order:

```
"user1", "AAPL", 10, 230.0
```

denotes user1 wants to trade 10 shares of "AAPL" stock with the price 230.0.

We organize one order information as an order object below.

```
class Order {
public:
    std::string user_id;
    std::string symbol;
    int quantity;
    double price;
    Order* next;

    Order(const std::string& uid, const std::string& sym, int qty, double
pr)
        : user_id(uid), symbol(sym), quantity(qty), price(pr),
next(nullptr) {}
};
```

This assignment asks to implement a **LinkedList** where **each node in the LinkedList is an Order object**. The LinkedList class includes methods for appending, inserting, removing, searching, and displaying orders, as the LinkedList operations we learned in the class. The LinkedList ADT is shown below, and the detailed explainations of these operations can be found in OrderLinkedList-stu.cpp.

```
class LinkedList {
private:
    Order* head;

public:
    LinkedList() : head(NULL) {}
    ~LinkedList();
    void display() const;
    void append(const string& user_id, const string& symbol, int quantity,
double price);
    bool insert(int index, const string& user_id, const string& symbol,
int quantity, double price);
    int getSize() const;
    bool search(const string& symbol) const;
    bool remove(int index);
};
```

The main function demonstrates adding orders to the list, searching for an order, removing an order, and displaying the list as below:

```
int main() {
    LinkedList orders;

    // Example usage
    orders.append("user1", "AAPL", 10, 230.0);
    orders.insert(0, "user1", "AAPL", 5, 235.0);
    orders.append("user1", "GOOGL", 8, 240.0);

    cout << "Orders in the list:" << endl;
    orders.display();

    cout << "Size of the list: " << orders.getSize() << endl;

    if (orders.search("AAPL")) {
        cout << "AAPL order found in the list." << endl;
    }

    if (orders.remove(1)) {
        cout << "Order removed from the list." << endl;
    }

    cout << "Orders after removing one item:" << endl;
    orders.display();

    return 0;
}
```

and the desired output is:

```
Orders in the list:
Order - User ID: user1, Symbol: AAPL, Quantity: 5, Price: 235
Order - User ID: user1, Symbol: AAPL, Quantity: 10, Price: 230
Order - User ID: user1, Symbol: GOOGL, Quantity: 8, Price: 240
Size of the list: 3
Symbol is found
AAPL order found in the list.
Order removed from the list.
Orders after removing one item:
Order - User ID: user1, Symbol: AAPL, Quantity: 5, Price: 235
Order - User ID: user1, Symbol: GOOGL, Quantity: 8, Price: 240
The Linked list has been deleted.
```