```
package com.jinglin;

public class Item {
    private String itemName;
    private double price;
    private int amount;

    public Item(String itemName, double price) {
        this.itemName = itemName;
        this.price = price;
        this.amount = 0;
    }

    public void storeAmount(int amount) {
        this.amount += amount;
    }

    public String getItemName() {
        return itemName;
    }

    public double getPrice() {
        return price;
    }

    public int getAmount() { return amount; }

    public void shipAmount(int shipAmount) {
        this.amount -= shipAmount;
    }

    public void receiveAmount(int receiveAmount){
        this.amount += receiveAmount;
    }
}
```

```
package com.jinglin;

import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

public class Main {

    private static ArrayList<WareHouse> wareHouses = new ArrayList<WareHouse>();

    public static void main(String[] args) throws IOException{
        File myFile = new File("c:/myFile.txt");
        Scanner input = new Scanner(myFile);
        PrintWriter output = new PrintWriter("c:/myOutput.txt");

        wareHouses.add(new WareHouse("New York"));
        wareHouses.add(new WareHouse("Miami"));
        wareHouses.add(new WareHouse("Los Angeles"));
        wareHouses.add(new WareHouse("Houston"));
        wareHouses.add(new WareHouse("Chicago"));

        while(input.hasNext()){
            String str = input.nextLine();
            String[] strArray = str.split("\t");
            for(String s: strArray){
                output.printf("%s  ", s);
            }
            output.println();

            WareHouse wareHouse = searchWareHouse(strArray[1]);
            if(wareHouse != null) {
                if (strArray[0].equals("s")) {
                    wareHouse.store(Integer.parseInt(strArray[2]),
                            Integer.parseInt(strArray[3]), Integer.parseInt(strArray[4]));
                    //method to print out updated warehouse
                    wareHouse.printWareHouseRecord(output);
                } else if (strArray[0].equals("o")) {
                    double priceOfOrder = 0;
                    double priceOfShipping = shipItemsNeededForOrder(strArray, wareHouse, output);

                    if(priceOfShipping != -1) {
                        ArrayList<Item> items = wareHouse.getItems();
                        for(int i = 0; i < items.size(); i++){
                            priceOfOrder += items.get(i).getPrice() *
                                    Integer.parseInt(strArray[i + 2]);
                            items.get(i).shipAmount(Integer.parseInt(strArray[i + 2]));
                        }
                        priceOfOrder += priceOfShipping;
                        output.printf("Price of Order: $%.2f", priceOfOrder);
                        output.println();
                        wareHouse.printWareHouseRecord(output);
                    }else{
                        output.println("Order Unfilled");
                        wareHouse.printWareHouseRecord(output);
                    }
                }
            }else{
                output.println("Error: Ware House not found.");
            }
            output.println();
            output.println("==========================================");
            output.println();
        }
        output.flush();
```

```
        System.out.println("The program has finished.");
        input.close();
        output.close();
    }

    public static WareHouse searchWareHouse(String name){
        for(WareHouse w: wareHouses){
            if(w.getWareHouseName().equals(name)){
                return w;
            }
        }
        return null;
    }

    public static double shipItemsNeededForOrder(String[] strArray, WareHouse wareHouse,
                                                 PrintWriter output){
        double priceOfShipping = 0;
        for(int i = 0; i < wareHouse.getItems().size(); i++){
            int amountInStore = wareHouse.getItems().get(i).getAmount();
            int amountNeeded = Integer.parseInt(strArray[i + 2]);
            if(amountInStore < amountNeeded){
                int amountForShipping = amountNeeded - amountInStore;
                WareHouse maxWareHouse = searchWareHouseWithMostAmount(i, amountForShipping,
                        wareHouse);
                if(maxWareHouse != null){
                    priceOfShipping += 0.1 * maxWareHouse.getItems().get(i).getPrice() *
                            amountForShipping;
                    maxWareHouse.getItems().get(i).shipAmount(amountForShipping);
                    wareHouse.getItems().get(i).receiveAmount(amountForShipping);
                    output.println(amountForShipping + " of " +
                            maxWareHouse.getItems().get(i).getItemName() +
                            " shipped from " + maxWareHouse.getWareHouseName() + " to " +
                            wareHouse.getWareHouseName());
                    ArrayList<Item> items = maxWareHouse.getItems();
                    output.printf("%s\t%d\t%d\t%d", maxWareHouse.getWareHouseName(),
                            items.get(0).getAmount(), items.get(1).getAmount(),
                            items.get(2).getAmount());
                    output.println();
                }else{
                    return -1;
                }
            }
        }
        return priceOfShipping;
    }

    public static WareHouse searchWareHouseWithMostAmount(int i, int amountForShipping,
                                                          WareHouse wareHouse){
        int max = 0;
        WareHouse maxWareHouse = null;
        for(WareHouse w: wareHouses){
            int amount = w.getItems().get(i).getAmount();
            if(w != wareHouse && amount >= amountForShipping && amount > max){
                maxWareHouse = w;
                max = amount;
            }
        }
        return maxWareHouse;
    }


}
```

```
package com.jinglin;

import java.io.PrintWriter;
import java.util.ArrayList;

public class WareHouse {
    private String wareHouseName;
    private ArrayList<Item> items = new ArrayList<Item>();

    public WareHouse(String wareHouseName){
        this.wareHouseName = wareHouseName;
        this.items.add(new Item("item 1", 2));
        this.items.add(new Item("item 2", 7));
        this.items.add(new Item("item 3", 8.5));
    }

    public String getWareHouseName() {
        return wareHouseName;
    }

    public ArrayList<Item> getItems() {
        return items;
    }

    public void store(int amount1, int amount2, int amount3){
        items.get(0).storeAmount(amount1);
        items.get(1).storeAmount(amount2);
        items.get(2).storeAmount(amount3);
    }

    public void printWareHouseRecord(PrintWriter output){
        output.printf("Updated: %s  %d  %d  %d",wareHouseName, items.get(0).getAmount(),
                items.get(1).getAmount(), items.get(2).getAmount());
        output.println();
    }


}
```