```cpp
///Marc Pfeiffer
///Assignment 2 - Warehouse Inventory System

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

ifstream dataIn("shipmentFile.txt");
ofstream dataOut("outfile.txt");

class Shipment{

public:

    friend ifstream& operator >> (ifstream& dataIn, Shipment& order);
    friend ofstream& operator << (ofstream& dataOut, Shipment& order);
    char getShipType(){return shipType;}
    string getWarehouseName(){return warehouseName;}
    int getAmountArr(int i){return *(amountArr+i);}

private:

    char shipType;
    string warehouseName;
    int amountArr[3];
};
/// Overloaded in-file Operator reads in data from each order
ifstream& operator >> (ifstream& dataIn, Shipment& order){

    dataIn>>order.shipType>>order.warehouseName;

    for(int i = 0 ; i<3 ; i++){
        dataIn>>order.amountArr[i];
    }

    return dataIn;
}

/// Overloaded out-file Operator outputs data from each order
ofstream& operator << (ofstream& dataOut, Shipment& order){

    dataOut<<order.shipType<<" "<<order.warehouseName<<" ";

    for(int i = 0 ; i<3 ; i++){
        dataOut<<order.amountArr[i]<<" ";
    }
    dataOut<<"\n\n";

    return dataOut;
}
/// Adds amounts of different objects up
/// returns nothing
void addAmounts(int, int[5][3], Shipment&) ;

///if an order was placed amounts are subtracted from warehouse
/// returns nothing

void subtractAmounts(int, int[5][3], Shipment&,double[3]);

/// finds the largest amount from all warehouses
/// returns largest
int findLargest(int, int& , int[5][3]);


```

```
int main()
{
    double priceArr[3] = {2.00, 7.00, 8.50};

    string places[5] = {"Chicago", "Houston", "L.Angl." , "Miami" ,"NewYork"};

    int house;
    int arr[5][3];

    for(int i = 0; i < 5; i++){
        for(int j = 0; j<3; j++){
            arr[i][j] = 0;
        }
    }

    Shipment order;

    ///loops till end of shipment file
    while(!dataIn.eof()){

         dataIn>>order;
         dataOut<<order;

        if(order.getWarehouseName()=="Chicago")
            house = 0;
        else if(order.getWarehouseName()=="Houston")
            house = 1;
        else if(order.getWarehouseName()=="LosAngeles")
            house =2;
        else if(order.getWarehouseName()=="Miami")
            house = 3;
        else
            house = 4;

        ///if shipment comes in
        if(order.getShipType()=='s'){
            addAmounts(house, arr, order);
        }

        /// if shipment is sent out
        if(order.getShipType()=='o'){
            subtractAmounts(house, arr, order, priceArr);
        }

        ///prints out current status of warehouses.
        for(int i = 0 ; i<5; i++){
            dataOut<<places[i];
            for(int j = 0 ; j<3; j++){
                dataOut<<"\t\t"<<arr[i][j];
            }
            dataOut<<endl;
        }
        dataOut<<endl;

    }
    return 0;
}

///adds shipment to inventory
void addAmounts(int houseNum, int arr[5][3], Shipment &obj) {

    for(int i = 0; i<3 ; i++){
        arr[houseNum][i] += obj.getAmountArr(i);
    }
 }

```

```
133  ///subtracts shipment from inventory
134  void subtractAmounts(int houseNum, int arr[5][3], Shipment &obj, double priceArr[3]
){
135      dataOut.setf(ios::fixed, ios::floatfield);
136      dataOut.precision(2);
137
138      double priceOfOrder =0;
139
140      for(int i = 0; i<3 ; i ++){
141
142          ///if selected warehouse has enough inventory
143          if(obj.getAmountArr(i) <= arr[houseNum][i]){
144
145              arr[houseNum][i] -= obj.getAmountArr(i) ;
146              priceOfOrder += (obj.getAmountArr(i) * priceArr[i]);
147          }
148
149          else{
150
151              int j=0;
152
153              ///if a different warehouse has proper inventory
154              if( findLargest(i, j, arr) >= obj.getAmountArr(i) ){
155
156                  arr[j][i] -= obj.getAmountArr(i) ;
157                  priceOfOrder += ((obj.getAmountArr(i) *priceArr[i])*1.10);
158                  dataOut<<"Item "<< i+1 <<" shipped from a different location, price
has changed.\n";
159              }
160
161              /// if no warehouses have proper inventory
162              else{
163
164                  dataOut <<"Order Unfilled - Item "<< i+1 <<" wasn't sent, Due to
lack of inventory.\n";
165              }
166          }
167      }
168
169      dataOut<<"Price of Order: $"<<priceOfOrder<<"\n\n";
170  }
171
172   ///finds largest
173   ///returns largest
174   int findLargest(int i , int &j, int arr[5][3]){
175
176      int largest = arr[0][i];
177
178      for(int k =1 ; k<5 ; k++){
179
180          if(arr[k][i] > largest){
181              largest = arr[k][i];
182              j=k;
183          }
184
185      }
186   return largest;
187   }
188
189
190
191
```