 An e-commerce company processes orders for customers if the quantity is in stock. The total value must not exceed their account balance. Given products, users with balances, and orders that users want to place, implement the functionality described.

Create the Product class and have it implement the IProduct interface .

This class should have the properties

* Id,
* Name,
* Price,
* ShippingCost.

It should have a constructor which takes these values as inputs and sets them as the object's properties.

Create the User class and have it implement the IUser interface.

This class should have the properties

* Id
* Name
* Balance
* Orders property of type List<(IProduct product, int quantity)>

It should have a constructor which takes id, name, and balance as inputs and sets them as the object's properties.

The Orders property should be initialized to a new instance of List<(IProduct product, int quantity)> in the constructor.

Create the Company class and have it implement the ICompany interface.

This class should have the properties

* Products of type List<(IProduct product, int quantity)>
* Users of type List<IUser>

It should have a constructor which initializes these properties to new instances of these types.

Implement the Make Order method in the Company class. It should

* Take a list of products and a user as input.
* Check that there are enough units available.
* Determine the highest shipping cost of all items ordered.
* The total cost is the sum of (unit price \* units) for each item ordered, plus the highest shipping cost.
* Check if the user has enough funds.

If these conditions are met, the method should update

* user's balance update
* product quantity on hand
* Add the products to the user's orderslist

Implement the AddProduct method.

Take a product and quantity as inputs

If the product exists in the Products list, increase the quantity on handotherwise, add the product to the Productslist

Implement the AddUser method. It should take a user as inputadd the user to the Users list.

Example
There are 2 products and 1 user. This user orders 3 Laptops and 1 Phone with 100 funds in the account.
*Quantity will

Constraints
- Product IDs are distinct for each product.
I Input Format For Custom Testing
The first line contains an integerExplanation
Looking at the order, user 1 orders 5 units of product 1 and 2 units of product2. The higher price for shipping i