C++ OOP – Regular Exam – 25 August 2024

2. Yachting

You're tasked with developing a program, which will read and analyze yacht data. The program uses class hierarchy to store the data. Your task is to study the code, the input and output data, and to complete the missing code and classes to achieve successfully running solution.

Input

The input consists of many of the following lines:

1. **"Yacht YachtName docking"**: this indicates that a new yacht, with name **YachtName** is docking in the port.
2. **"Yacht YachtName undocking"**: the yacht with the name **YachtName** is undocking from the port.
3. **"Supply SupplyType SupplyNumber for YachtName arriving"**: new supply of type **SupplyType** and quantity **SupplyNumber** is **YachtName** at the dock.
4. **"Passenger PassengerName arrived for YachtName"**: a new passenger with name **PassengerName** is arriving for the yacht **YachtName**.
5. **"End"**: the last line of the input. Processing stops, no more lines will follow.

Note: **YachtName**, **SupplyType** and **PassengerName** will not contain space or other stream divisor symbol. **SupplyNumber** will always be a valid integer.

Output

For each input line, the program produces a output, depending on the result of the processing of the libe. After reading each command, the software analyzes the current data, updates it, and outputs information, depending on the command and its result:

1. For command **(1) "yacht docking"**:
    a. If there is already a yacht with the same name on dock, the output should be: **"Error: Yacht YachtName already on dock!"**
    b. If there's no yacht with the same name, the output should be: **"Yacht YachtName successfully docked."**
2. For command **(2) "yacht undocking"**:
    a. If there's already a docked yacht with the same name, the output should be one line with all yacht data: **"Yacht YachtName successfully undocked with passengers [Passenger names in alphabetical order, separted by comma] and supplies [Supplies in format SupplyName:TotalQuantity, separated by comma]. "**. If any of the two lists are empty, the output should be **"[empty]"**.
    b. If there's no such yacht on dock, the output should be **"Error: Yacht YachtName is not on dock!"**
3. For command **(3) "supply arriving"**
    a. If there's no yacht, the error message **"Error: Yacht YachtName is not on dock!"**
    b. If there's yacht, the output should be **"Supply SupplyName added to yacht YachtName. New quantity: <Quantity>."**, where **Quantity** is the new quantity of that supply in the yacht.

c. **Please note:** there might be multiple additions of the same supply, in which case it adds up in the yacht's storage. For example, the commands:
   i. **`Supply Booze 30 for Sunrise90 arriving`**, would print **`"Supply Booze added to yacht Sunrise90. New quantity: 30."`**
   Then if sometimes later there's another input:
   ii. **`Supply Booze 55 for Sunrise90 arriving`**, then output should be **`"Supply Booze added to yacht Sunrise90. New quantity: 85."`**, as **`Sunrise90`** already had **`Booze`** in it, and **`55`** must be added to the already existing **`30`**.

4. For command **`(4) "passenger arriving"`**:
   a. If a passenger with the same name already exists in the crew, the output should be **`"Error: passenger PassengerName already exists in YachtName crew!"`**
   b. If there's no already existing passenger, the output should be **`"Passenger PassengerName added to crew. New crew: [<an alphabetically sorted list of all crew members>]."`**
5. For command **`(5) "End"`**:
   a. If there are any yachts on dock, the system should print the following line for each docked yacht: **`"Yacht YachtName remains on dock with passengers <Passenger names in alphabetical order, separted by comma> and supplies <Supplies in format SupplyName:TotalQuantity, separated by comma>."`**
   b. If there are no yachts on dock, the output should be **`"Dock empty."`**

Examples

Input	Output
`Yacht Dreamcatcher docking`	`Yacht Dreamcatcher successfully docked.`
`Supply Fuel 200 for Dreamcatcher arriving`	`Supply Fuel added to yacht Dreamcatcher. New quantity: 200.`
`Passenger Emma arrived for Dreamcatcher`	`Passenger Emma added to crew. New crew: [Emma].`
`Supply Fuel 50 for Dreamcatcher arriving`	`Supply Fuel added to yacht Dreamcatcher. New quantity: 250.`
`Passenger Michael arrived for Dreamcatcher`	`Passenger Michael added to crew. New crew: [Emma, Michael].`
`Yacht Dreamcatcher undocking`	`Yacht Dreamcatcher successfully undocked with passengers [Emma, Michael] and supplies [Fuel:250].`
`End`	`Dock empty.`

Input	Output
`Yacht SeaBreeze docking`	`Yacht SeaBreeze successfully docked.`
`Yacht OceanQueen docking`	`Yacht OceanQueen successfully docked.`
`Yacht SeaBreeze docking`	`Error: Yacht SeaBreeze already on dock!`
`Yacht MorningStar undocking`	`Error: Yacht MorningStar is not on dock!`

Yacht OceanQueen undocking	Yacht OceanQueen successfully undocked with passengers [empty] and supplies [empty].
Supply Fuel 100 for MorningStar arriving	Error: Yacht MorningStar is not on dock!
Supply Booze 30 for SeaBreeze arriving	Supply Booze added to yacht SeaBreeze. New quantity: 30.
Supply Booze 55 for SeaBreeze arriving	Supply Booze added to yacht SeaBreeze. New quantity: 85.
Passenger John arrived for SeaBreeze	Passenger John added to crew. New crew: [John].
Passenger Alice arrived for SeaBreeze	Passenger Alice added to crew. New crew: [Alice, John].
Passenger John arrived for SeaBreeze	Error: passenger John already exists in SeaBreeze crew!
Supply Water 20 for SeaBreeze arriving	Supply Water added to yacht SeaBreeze. New quantity: 20.
End	Yacht SeaBreeze remains on dock with passengers [Alice, John] and supplies [Booze:85, Water:20].