

Managing Fruits

By: Ricardo & Billy



Deskripsi

Seorang pedagang buah bernama Pak Chanek ingin memodernisasi cara ia mengelola inventaris buah di tokonya. Ia membutuhkan sebuah program sederhana untuk mencatat stok buah-buahannya secara real-time.

Program ini harus bisa menangani dua jenis perintah:

- **ADD FRUIT_NAME QUANTITY:** Perintah ini digunakan untuk menambah stok buah. Jika buah dengan **FRUIT_NAME** tersebut sudah ada di dalam inventaris, jumlah stoknya akan ditambahkan. Jika belum ada, buah tersebut akan ditambahkan sebagai entri baru dengan stok sebanyak jumlah.
- **SELL FRUIT_NAME QUANTITY:** Perintah ini digunakan untuk menjual buah. Sebelum menjual, program harus melakukan validasi:
 - Jika buah dengan **FRUIT_NAME** tersebut tidak ada di dalam inventaris, penjualan gagal.
 - Jika buah tersebut ada, namun jumlah stoknya kurang dari jumlah yang ingin dijual, penjualan juga gagal.
 - Jika buah tersebut ada dan stoknya mencukupi, penjualan berhasil dan stok buah tersebut akan dikurangi.

Buatlah program untuk memproses serangkaian perintah tersebut dan memberikan respons yang sesuai untuk setiap perintah **SELL**.

Hint: `std::map`

Format Masukan

Baris pertama berisi sebuah bilangan bulat **N**, yaitu jumlah total perintah yang akan diproses.

- **N** baris berikutnya masing-masing berisi sebuah perintah dalam format: **PERINTAH FRUIT_NAME QUANTITY**
- **COMMAND** adalah sebuah string, "**ADD**" atau "**SELL**".
- **FRUIT_NAME** adalah sebuah string yang merepresentasikan nama-nama buah yang dijual.
- **JUMLAH** adalah sebuah bilangan bulat yang merepresentasikan jumlah buah.

Format Keluaran

Untuk setiap perintah **SELL**, cetak satu baris string yang merupakan hasil dari transaksi tersebut.

- **SUCCESS**: Jika penjualan berhasil.
- **item not found**: Jika buah yang ingin dijual tidak ada di dalam inventaris.
- **not enough stock**: Jika stok buah tidak mencukupi.

Perintah **ADD** tidak menghasilkan keluaran apa pun.

Batasan

$1 \leq N, \text{QUANTITY} \leq 2^{16}$

FRUIT_NAME terdiri dari 1 hingga 15 karakter huruf kecil (a-z).

Contoh

Sample Input 0
9 ADD mango 20 ADD rambutan 15 SELL mango 5 SELL durian 1 ADD mango 10 SELL rambutan 20 SELL mango 25 ADD salak 30 SELL mango 1
Sample Output 0

SUCCESS
item not found
not enough stock
SUCCESS
not enough stock

Penjelasan

Stok Awal (map): {}

1. **ADD** mango 20
 - a. map menjadi: {"mango": 20}
2. **ADD** rambutan 15
 - a. map menjadi: {"mango": 20, "rambutan": 15}
3. **SELL** mango 5
 - a. Cek: mango ada, stok (20) > 5. Berhasil.
 - b. map menjadi: {"mango": 15, "rambutan": 15}
 - c. Output: **SUCCESS**
4. **SELL** durian 1
 - a. Cek: durian tidak ditemukan di dalam map. Gagal.
 - b. Output: **item not found**
5. **ADD** mango 10
 - a. Stok mango menjadi 15 + 10 = 25.
 - b. map menjadi: {"mango": 25, "rambutan": 15}
6. **SELL** rambutan 20
 - a. Cek: rambutan ada, tapi stok (15) < 20. Gagal.
 - b. Output: **not enough stock**
7. **SELL** mango 25
 - a. Cek: mango ada, stok (25) >= 25. Berhasil.
 - b. Stok mango menjadi 25 - 25 = 0.
 - c. map menjadi: {"mango": 0, "rambutan": 15}
 - d. Output: **SUCCESS**
8. **ADD** salak 30
 - a. map menjadi: {"mango": 0, "rambutan": 15, "salak": 30}
9. **SELL** mango 1
 - a. Cek: mango ada, tapi stok (0) < 1. Gagal.
 - b. Output: **not enough stock**

Managing Fruits

By: Ricardo & Billy



Description

A fruit merchant named Chanek wants to modernize the way he manages his fruit inventory. He needs a simple program to track his fruit stocks in real-time.

This program must be able to handle two types of commands:

- **ADD FRUIT_NAME QUANTITY:** This command is used to add fruit to the fruit stock. If a fruit with the name **FRUIT_NAME** already exists in the inventory, its stock count will be increased. If it does not exist, the fruit will be added as a new entry with a stock of **QUANTITY**.
- **SELL FRUIT_NAME QUANTITY:** This command is used to sell fruit. Before selling, the program must perform a validation:
 - If the fruit with the name **FRUIT_NAME** does not exist in the inventory, the sale fails.
 - If the fruit exists, but its stock count is less than the **QUANTITY** to be sold, the sale also fails.
 - If the fruit exists and the stock is sufficient, the sale is successful, and the stock of that fruit will be decreased.

Create a program to process a series of these commands and provide the appropriate response for each **SELL** command.

Hint: std::map

Input Format

The first line contains an integer **N**, which is the total number of commands to be processed.

The next **N** lines each contain a command in the format: **COMMAND FRUIT_NAME QUANTITY**.

- **COMMAND** is a string, either "**ADD**" or "**SELL**".
- **FRUIT_NAME** is a string representing the name of a fruit.
- **QUANTITY** is an integer representing the quantity of the fruit.

Output Format

For each **SELL** command, print a single line string which is the result of the transaction.

- **SUCCESS**: If the sale is successful.
- **item not found**: If the fruit to be sold is not in the inventory.
- **not enough stock**: If the fruit stock is insufficient.

The **ADD** command produces no output.

Constraints

$1 \leq N, \text{QUANTITY} \leq 2^{16}$

FRUIT_NAME consists of 1 to 15 lowercase letters (a-z).

Example

Sample Input 0
9 ADD mango 20 ADD rambutan 15 SELL mango 5 SELL durian 1 ADD mango 10 SELL rambutan 20 SELL mango 25 ADD salak 30 SELL mango 1
Sample Output 0
SUCCESS item not found not enough stock SUCCESS

not enough stock

Explanation

Initial Stock (map): {}

1. **ADD** mango 20
 - a. the map becomes: {"mango": 20}
2. **ADD** rambutan 15
 - a. the map becomes: {"mango": 20, "rambutan": 15}
3. **SELL** mango 5
 - a. Check: mango exists, stock (20) \geq 5. Successful.
 - b. the map becomes: {"mango": 15, "rambutan": 15}
 - c. Output: **SUCCESS**
4. **SELL** durian 1
 - a. Check: durian is not found in the map. Fails.
 - b. Output: **item not found**
5. **ADD** mango 10
 - a. mango stock becomes $15 + 10 = 25$.
 - b. the map becomes: {"mango": 25, "rambutan": 15}
6. **SELL** rambutan 20
 - a. Check: rambutan exists, but stock (15) $<$ 20. Fails.
 - b. Output: **not enough stock**
7. **SELL** mango 25
 - a. Check: mango exists, stock (25) \geq 25. Successful.
 - b. mango stock becomes $25 - 25 = 0$.
 - c. the map becomes: {"mango": 0, "rambutan": 15}
 - d. Output: **SUCCESS**
8. **ADD** salak 30
 - a. the map becomes: {"mango": 0, "rambutan": 15, "salak": 30}
9. **SELL** mango 1
 - a. Check: mango exists, but stock (0) $<$ 1. Fails.
 - b. Output: **not enough stock**