

# Final Exam of Algorithms and Data Structures

## 2023-2024

An industrial pastry shop wants to improve its order management system and has commissioned you to develop software that simulates the operation of the pastry shop. The entire simulation takes place in discrete time. It is assumed that, upon execution of each received input command, an instant of time passes. The simulation starts at time 0. The simulation must consider the following elements:

- **The ingredients** of the pastries, each identified by its *name*, consisting of a sequence of characters.
- **The set of recipes** offered by the pastry shop, also identified by a *name*. Each recipe uses different *quantities* of each required ingredient (indicated as an integer number, in grams).
- **The warehouse** of pastry ingredients, which stores all the ingredients used. The warehouse is *restocked* with new *batches* of ingredients based on a supply schedule with the supplier. Each batch is characterized by a *quantity* (always in grams) and an *expiration date*, indicated as the time instant from which the batch is expired.
- **Customers** of the pastry shop place *orders* for one or more pastries through an online platform or by phone. In any case, the pastry shop immediately proceeds to prepare the ordered pastries. Advanced (and expensive) machines that prepare the pastries are so fast that the preparation of an arbitrary number of pastries occurs in a single instant of the simulation. The necessary ingredients for the preparation are taken from the warehouse, always prioritizing batches with the closest expiration date. If there are not enough ingredients available to allow the preparation of an *entire* order, the order is placed *on hold*. An arbitrary number of orders can be on hold. The pastry shop proceeds to prepare any subsequent orders. At each restocking, the pastry shop evaluates whether it is possible, with the received ingredients, to prepare orders currently on hold. If this is the case, it prepares them

in the same time instant. Orders on hold are processed in *chronological order* of receipt.

- **Periodically, the courier** arrives at the pastry shop to pick up ready orders. Upon arrival, the orders to be loaded are chosen in chronological order of arrival. The loading process stops as soon as an order is encountered that exceeds the remaining *capacity* (in grams) of the truck. It is assumed that the weight of each prepared pastry is equal to the sum of the quantities in grams of each ingredient. Each order is always fully loaded. Once the orders are chosen, the pastry shop proceeds to load them in *descending order* of weight. If weights are equal, orders are loaded in chronological order of arrival.

It is assumed that all quantities are whole and greater than zero, regardless of the unit of measure. Names, whether of ingredients or recipes, are defined over the alphabet {a, ..., z, A, ..., Z, -, 0, ..., 9} and are at most 255 characters long. The simulation ends after reading the last command.

The input text file begins with a line containing two integers: the *courier periodicity* and its *capacity*. A sequence of commands follows, one per line, in the following format. All positive or zero whole numbers are codifiable in 32 bits.

- **aggiungi\_ricetta** (*recipe\_name*) (*ingredient\_name*) (*quantity*) ...  
Example: `aggiungi_ricetta meringhe_della_prozia zucchero 100 albumi 100`  
Adds a recipe to the catalog. The number of (*ingredient\_name*) (*quantity*) pairs is arbitrary. If a recipe with the same name already exists, it is ignored. Expected output: **added** or **ignored**.
- **rimuovi\_ricetta** (*recipe\_name*)  
Example: `rimuovi_ricetta cannoncini`  
Removes a recipe from the catalog. It has no effect if the recipe does not exist, or if there are still orders related to it that have not yet been shipped. Expected output: **removed** or **orders pending on this recipe**.
- **rifornimento** (*ingredient\_name*) (*quantity*) (*expiration*) ...  
Example: `rifornimento zucchero 200 150 farina 1000 220`  
The pastry shop is restocked with a set of ingredient batches. The number of batches is arbitrary. Expected output: **restocked**.
- **ordine** (*recipe\_name*) (*number\_of\_items*)  
Example: `ordine torta_paradiso 36`

Places an order of (*number\_of\_items*) cakes with recipe (*recipe\_name*).  
Expected output: **accepted** or **rejected** if no recipe exists with the specified name.

In addition to the above outputs, the program prints the orders contained in the courier's truck as a sequence of triples (*arrival\_time*, *recipe\_name*, *number\_of\_items*), one per line, in loading order. Given a courier with periodicity  $n$ , the printout occurs before processing commands at time  $kn$  with  $k \in \{1, 2, \dots\}$ . If the truck is empty, the message **empty truck** is printed.

## Example

The following is an example illustrating the expected operation of the program. The *time* column indicates the current time instant when each command is executed.

<i>t</i>	Ingresso	Uscita	Commento
—	5 325		Configurazione del corriere
0	aggiungi_ricetta torta farina 50 uova 10 zucchero 20	aggiunta	Aggiunta ricette
1	aggiungi_ricetta ciambella farina 20 uova 5 burro 2	aggiunta	
2	aggiungi_ricetta profiterole farina 10 uova 2 latte 3 zucchero 3 cioccolato 4	aggiunta	
3	rimuovi_ricetta sfogliatella	non presente	La ricetta non esiste
4	rifornimento farina 100 10 uova 100 10 zucchero 100 10 burro 100 10 latte 100 10 cioccolato 100 10	rifornito	
5		camioncino vuoto	Prima spedizione, al tempo 5, prima di leggere il prossimo comando
	ordine ciambella 6	accettato	Non c'è abbastanza farina, ordine in attesa
6	ordine profiterole 3	accettato	Ordine preparato e subito messo in coda per la spedizione
7	rimuovi_ricetta profiterole	ordini in sospeso	I <i>profiterole</i> sono in coda per la spedizione
8	aggiungi_ricetta pane_dolce farina 1 zucchero 1 uova 1	aggiunta	
9	ordine ciambella 3	accettato	Ordine preparato e subito messo in coda per la spedizione
10		9 ciambella 3 6 profiterole 3	L'ordine delle ciambelle è più pesante (81g vs. 66g) e viene caricato per primo
	ordine torta 1	accettato	La dispensa è vuota perché i lotti dell'ultimo rifornimento sono scaduti: l'ordine va in attesa.
11	rifornimento farina 100 15 farina 50 13 uova 45 20 zucchero 20 20 burro 15 20	rifornito	L'ordine delle ciambelle al tempo 5 viene preparato perché è in attesa da più tempo, la torta ordinata al tempo 10 resta in attesa
12	rifornimento farina 100 15 uova 7 15 zucchero 25 15 latte 5 15 cioccolato 5 15	rifornito	Questo terzo rifornimento sblocca l'ordine della torta
13	ordine torta 1	accettato	
14	ordine profiterole 1	accettato	Torta e <i>profiterole</i> vengono subito preparati
15		5 ciambella 6 10 torta 1 13 torta 1	I <i>profiterole</i> restano in coda per la spedizione perché non ci stanno sul camioncino e sono l'ordine più recente. Le due torte hanno lo stesso peso, quindi vengono caricate in ordine cronologico. Quello che resta dei lotti con scadenza a tempo 15 viene tolto dalla dispensa.

Figure 1: Example

At the end of execution, 3g of butter and 2g of sugar remain unused. The profiterole order at time 14 is never shipped.