

Prova finale di algoritmi e strutture dati 2023-2024

Una pasticceria industriale vuole migliorare il proprio sistema di gestione degli ordini, e vi incarica di sviluppare un software che simuli il funzionamento della pasticceria. L'intera simulazione avviene a tempo discreto. Si assuma che, a valle dell'esecuzione di ogni comando ricevuto in ingresso, trascorra un istante di tempo. La simulazione ha inizio al tempo 0. Nella simulazione si dovranno considerare i seguenti elementi:

- Gli **ingredienti** dei dolci, ognuno identificato dal suo *nome*, costituito da una sequenza di caratteri.
- L'insieme delle **ricette** offerte dalla pasticceria, identificate anch'esse da un *nome*. Ogni ricetta utilizza diverse *quantità* di ciascun ingrediente necessario (indicate da un numero intero, in grammi).
- Il **magazzino** degli ingredienti della pasticceria, che stocca ogni ingrediente utilizzato. Il magazzino viene **rifornito** da nuovi **lotti** di ingredienti, in base a una pianificazione stabilita dal fornitore. Ogni **lotto** è caratterizzato da una *quantità* (sempre in grammi) e da una *data di scadenza*, indicata dal numero dell'istante di tempo a partire dal quale il lotto è scaduto.
- I clienti della pasticceria effettuano **ordini** di uno o più dolci tramite una piattaforma online o per telefono. In ogni caso, la pasticceria procede subito a preparare i dolci ordinati. Gli avanzati (e costosi) macchinari che preparano i dolci sono così veloci che si può assumere che la preparazione di un numero arbitrario di dolci avvenga in un singolo istante della simulazione. Gli ingredienti necessari per ogni preparazione vengono prelevati dal magazzino privilegiando sempre i lotti con la scadenza più prossima. Se non sono disponibili ingredienti a sufficienza da consentire la preparazione *per intero* di un ordine, esso viene messo in *attesa*. È possibile avere un numero arbitrario di ordini in attesa. La pasticceria procede a preparare eventuali ordini successivi. Ad ogni rifornimento, la pasticceria valuta se è possibile, con gli ingredienti ricevuti, preparare ordini attualmente in attesa. Se questo è il caso, li prepara nello stesso istante di tempo. **Gli ordini in attesa vengono smaltiti in ordine cronologico di arrivo dell'ordine.**
- Periodicamente, il **corriere** si reca dalla pasticceria a ritirare gli ordini pronti. All'arrivo del corriere, gli **ordini da caricare** vengono **scelti in ordine cronologico di arrivo**. Il processo si ferma appena viene **incontrato un ordine che supera la capienza rimasta (in grammi) sul camioncino**. Si assuma che il peso di ogni dolce preparato sia uguale alla somma delle quantità in grammi di ciascun ingrediente. Ogni ordine viene sempre caricato nella sua interezza. Scelti gli ordini, la pasticceria procede a caricarli in ordine di peso decrescente. A parità di peso, gli ordini vanno caricati in ordine cronologico di arrivo.

Si assuma che tutte le quantità siano intere e maggiori di zero, indipendentemente dall'unità di misura. I nomi, siano essi di ingredienti o ricette, sono definiti sull'alfabeto $\{a, \dots, z, A, \dots, Z, _ \}$ e sono lunghi fino a 255 caratteri. La simulazione termina in seguito alla lettura dell'ultimo comando.

Il file testuale in ingresso inizia con una riga contenente due interi: la periodicità del corriere e la sua capienza. Seguono una sequenza di comandi, uno per riga, con il seguente formato. Tutti i valori interi positivi o nulli sono codificabili in 32 bit.

- **aggiungi_ricetta** $\langle nome_ricetta \rangle \langle nome_ingrediente \rangle \langle quantità \rangle \dots$
Esempio: `aggiungi_ricetta meringhe_della_prozia zucchero 100 albumi 100`
Aggiunge una ricetta al catalogo. Il numero di coppie ($\langle nome_ingrediente \rangle \langle quantità \rangle$) è arbitrario. Se una ricetta con lo stesso nome è già presente, viene ignorato.
Stampa attesa come risposta: **aggiunta** oppure **ignorato**.
- **rimuovi_ricetta** $\langle nome_ricetta \rangle$
Esempio: `rimuovi_ricetta cannoncini`
Rimuove una ricetta dal catalogo. Non ha effetto se la ricetta non è presente, oppure ci sono ordini relativi ad essa non ancora spediti.
Stampa attesa come risposta: **rimossa**, **ordini in sospeso** oppure **non presente**.
- **rifornimento** $\langle nome_ingrediente \rangle \langle quantità \rangle \langle scadenza \rangle \dots$
Esempio: `rifornimento zucchero 200 150 farina 1000 220`
La pasticceria viene rifornita di un insieme di lotti, uno per ingrediente. Il numero di lotti è arbitrario.
Stampa attesa come risposta: **rifornito**.
- **ordine** $\langle nome_ricetta \rangle \langle numero_elementi_ordinati \rangle$
Esempio: `ordine torta_paradiso 36`
Effettua un ordine di $\langle numero_elementi_ordinati \rangle$ dolci con ricetta $\langle nome_ricetta \rangle$.
Stampa attesa come risposta: **accettato** oppure **rifiutato** se non esiste nessuna ricetta col nome specificato.

In aggiunta alle stampe sopraelencate, il programma stampa gli ordini contenuti nel camioncino del corriere come una sequenza di triple $\langle istante_di_arrivo_ordine \rangle \langle nome_ricetta \rangle \langle numero_elementi_ordinati \rangle$, una per riga, in ordine di caricamento. Dato un corriere di periodicità n , la stampa è effettuata prima di gestire i comandi al tempo kn con $k \in 1, 2, \dots$. Se il camioncino è vuoto, viene stampato il messaggio **camioncino vuoto.**

Esempio

Di seguito un esempio che illustra il funzionamento atteso del programma. La colonna *tempo* indica l'istante di tempo attuale quando viene eseguito ogni comando.

<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.

A fine esecuzione rimangono inutilizzati 3g di burro e 2g di zucchero. L'ordine dei *profiterole* al tempo 14 non viene mai spedito.