



Informații utile:

Enunțul problemei este prezentat mai jos.
Exemple intrare / ieșire + referințe vizuale: **IO**

Enunț:

Un supermarket online își dorește extinderea activității prin înființarea unui nou departament, denumit "Fresh Corner", care să poată oferi consumatorilor atât legume și fructe proaspete, direct de la producători autohtoni, cât și o eventuală producție proprie a unor preparate conexe (salate de fructe și legume, smoothies, etc).

Pentru a putea menține la un nivel optim noțiunea de "proaspăt" (aka Fresh), se propune implementarea unui nou modul în soluția existentă, dedicat managementului stocurilor și vânzărilor, atât prin prioritizarea vânzărilor produselor care se apropie de pragul de expirare cât și eliminarea din stoc a produselor considerate expirate.

Nu în ultimul rând, se dorește monitorizarea procesului, pentru a se asigura că este unul profitabil.

Input:

Se dă ca input un set de date care reflectă stocul, prețul, cantitatea și momentul de expirare cât și o listă cu cererile de cumpărare (data și numărul de produse din purchase order):

Exemplu input:

```
5
pere 1.00 50 7
pere 1.00 50 24
mere 0.50 70 11
mere 0.50 90 20
mere 0.50 40 27
4
2 1
pere 1.20 30
...
1
salata_fructe 2 mere 2 pere 1
```

Format input:

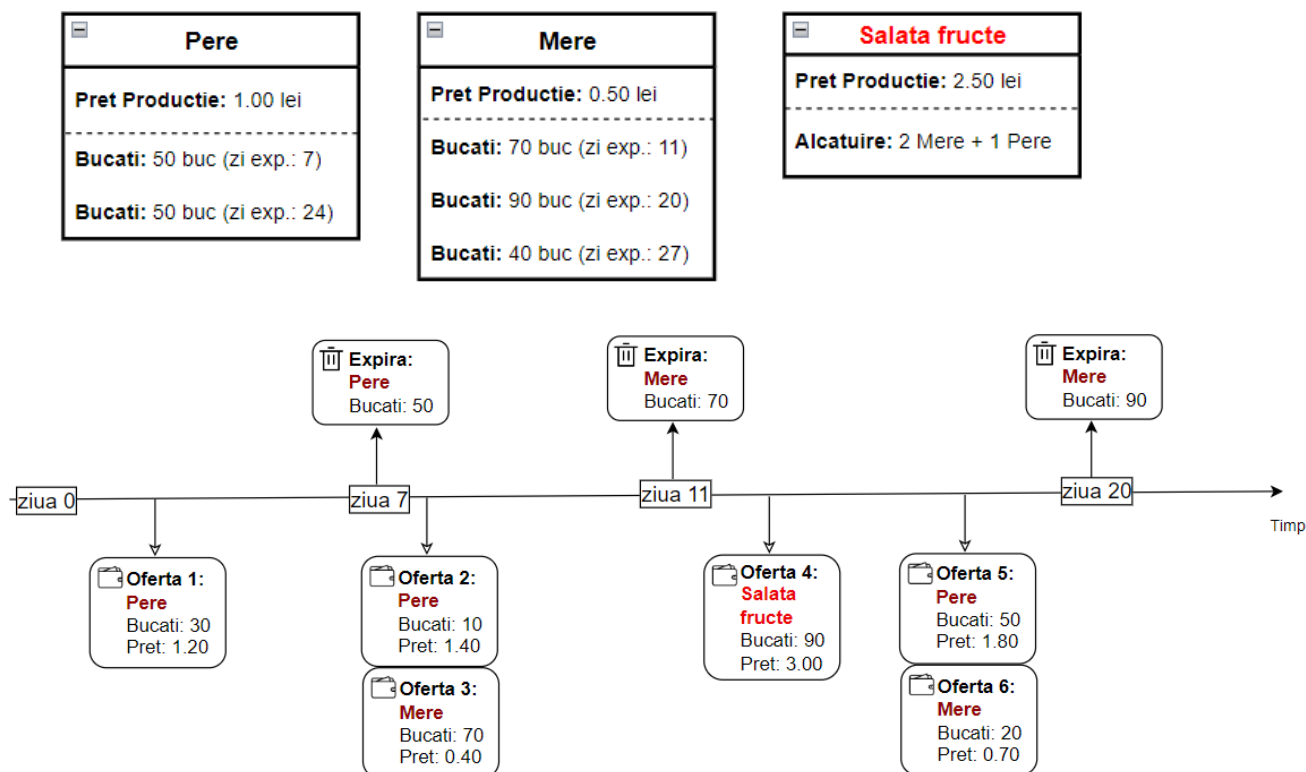
```
<Număr produse stoc>
<Produs> <Preț Producție> <Cantitate> <Dată Expirare>

<Număr Oferte Vânzare>
<Dată Ofertă> <Nr Produse>
<Produs> <Preț Ofertă> <Cantitate Ofertă>

...
<Număr Produse Preparate>
<Produs> <Nr Produse> <Produs> <Cantitate> <Produs> <Cantitate>
```

Reguli generale:

- Stocul are o cantitate finită.
- Ofertele trebuie tratate în regim FIFO (primul venit, primul servit).
- Produsele trebuie prioritizate după data de expirare.
- Produsele nu trebuie să fie expirate la vânzare.
- Se pot vinde cantități fracționare.
- Dacă oferta conține mai multe bucăți decât stocul, aceasta va fi acceptată cu numărul de bucăți disponibile.
- Produsele pot apărea de mai multe ori în stoc, cu zile de expirare diferite. Se garantează același preț de producție pentru acestea.
- Nu se acceptă oferta dacă prețul din ofertă este mai mic strict decât prețul de producție.



Să se creeze un algoritm care va fi folosit ulterior în soluția existentă și care să aibă ca output ofertele acceptate (Produs + Cantitate) și Total Incasări, respectând cerințele:

Cerința a:

Nu se consideră produsele preparate. Dacă produsul din ofertă nu există în stoc, nu se consideră.

Exemplu output:

pere 30
pere 10
pere 40
mere 20
136

Format output:

<Produs> <Cantitate>

<Total incasări>

Cerința b:

Introducem ideea de produse preparate (e.g. compuse). Un produs compus poate fi alcătuit din alte produse simple sau compuse, dându-se cantitatea necesară de compunere.

Pentru a putea vinde un produs preparat, toate componentele lui trebuie să fie disponibile. Cantitatea maximă de vânzare este dată de cantitatea de componente disponibile.

Exemplu output:

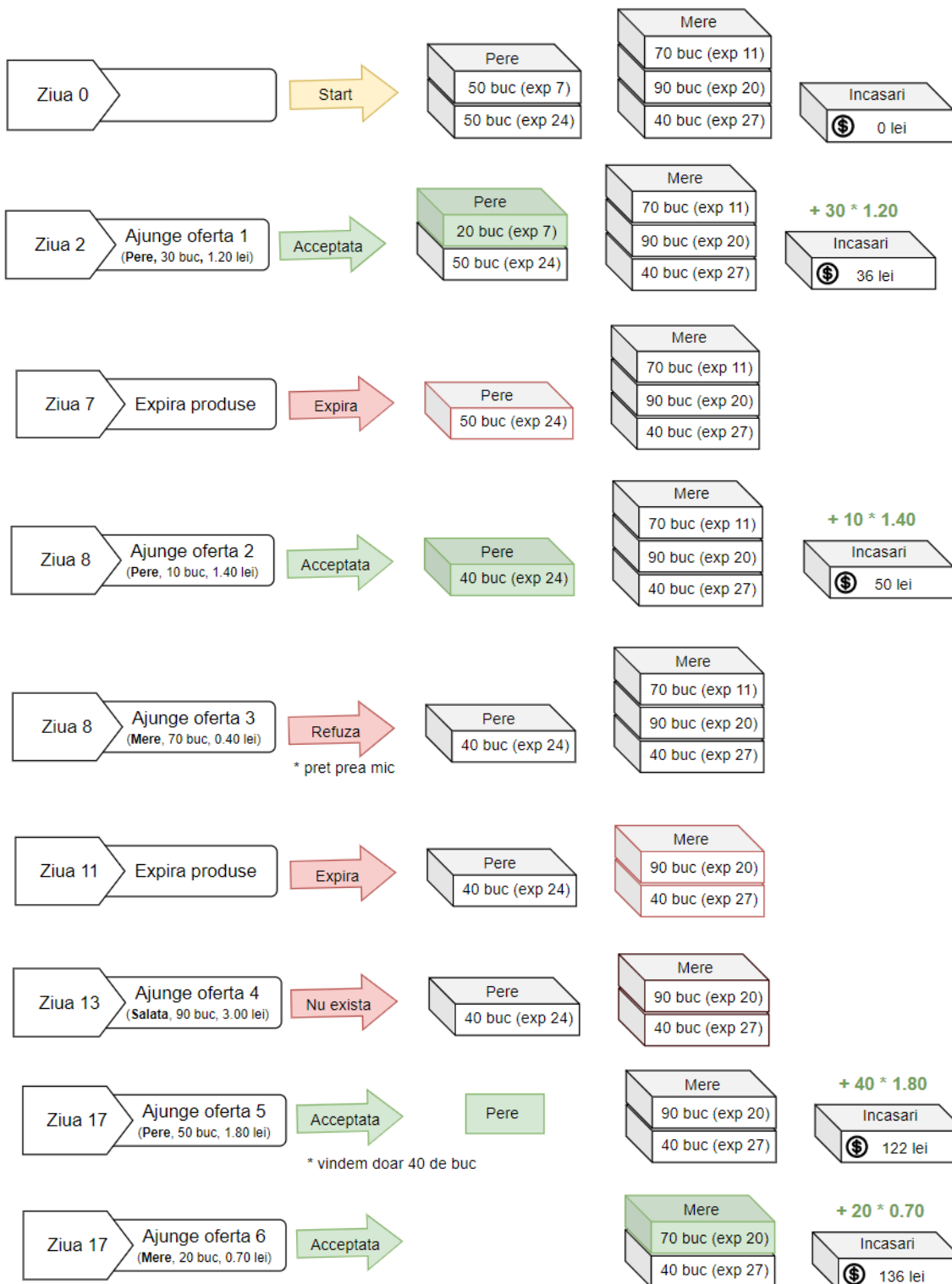
pere 30
pere 10
salata_fructe 40
mere 20
184

Format output:

<Produs> <Cantitate>

<Total incasări>

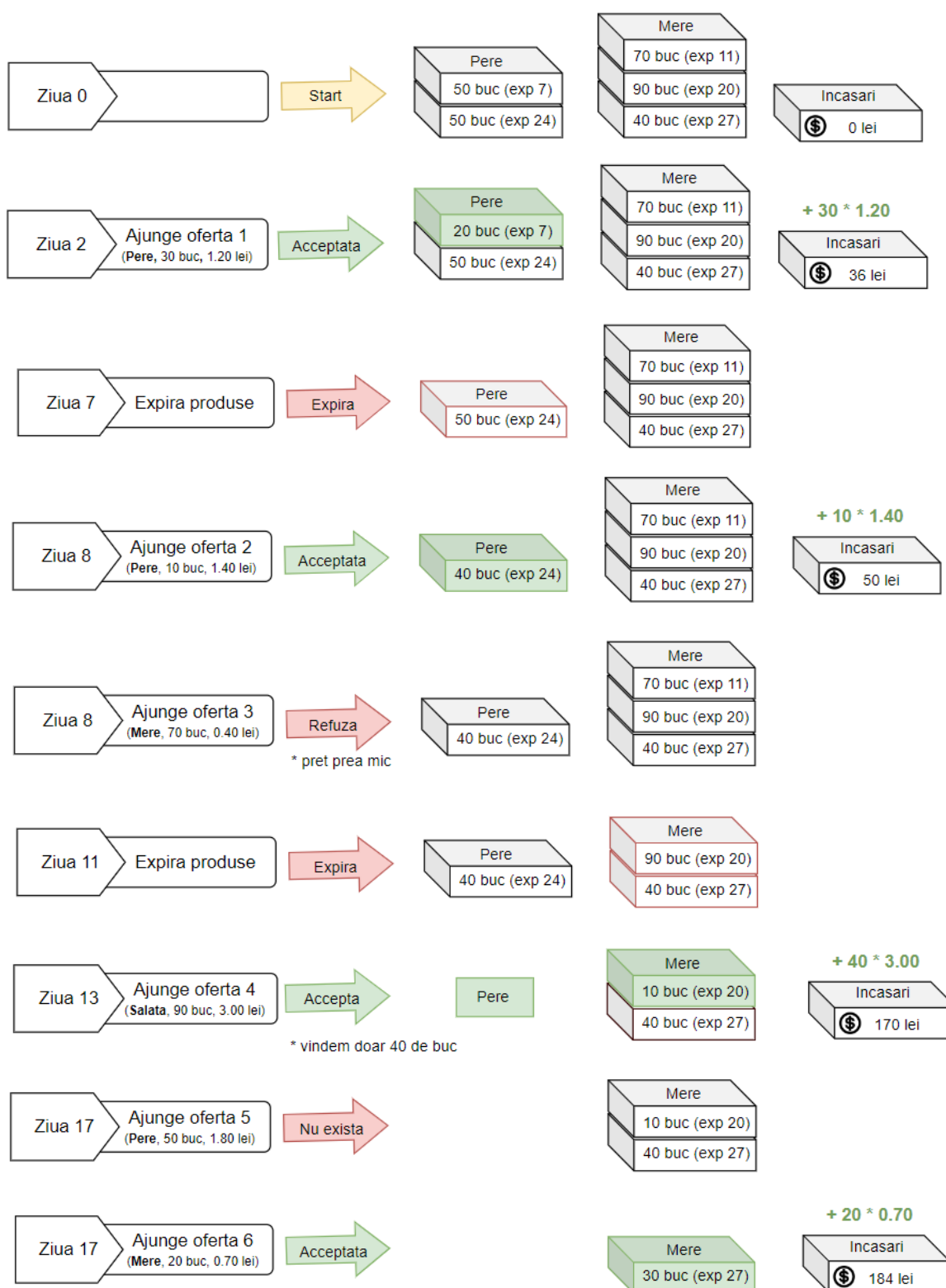
Referințe vizuale (Cerința a):



A se observa:

- Odată expirate, produsele ies din stoc.
- Dacă oferta este mai mica decât prețul de producție, aceasta se refuză.
- Produsele compuse nu sunt considerate în acest punct.
- Dacă oferta solicită mai multe produse decât cele disponibile, aceasta va fi acceptată cu cantitatea disponibilă.

Referințe vizuale (Cerința b):



A se observa:

- In acest punct, oferta cu *salată* va fi considerată.
- Deși existau suficiente *mere* în stoc pentru a crea un numar mai mare de *salate*, cantitatea de produse vândute este dată de numărul de *pere*, stocul acestora fiind redus.