

## Feladat

Egy raktár árukészlete a L nevű egyirányú, fejelemes listában (H1L) van tárolva. Egy listaelem tartalmazza az áru kódját, és a raktárban lévő mennyiséget (darabszám). A lista árukód szerint szigorúan monoton növekvően rendezett. Egy hasonló LM pointerű listában érkezik az aznapi kiszállítás: melyik termékből (kódja adott), mennyit vittek el a raktárból. Ez a lista is termékkód szerint szigorúan monoton növekvő. Aktualizáljuk a L listában található raktárkészletet a következők szerint:

- Ha a termék mindkét listában szerepel, akkor csökkentsük a raktárkészletet. Ha több lenne a kiviteli mennyiség, mint a raktárkészlet, jelezzünk hibát és ne módosítsuk a mennyiséget.
- Ha olyan árura érkezik kiviteli, amelyik nem szerepel a listában, szintén jelezzünk hibát.
- Hibajelzés után az algoritmus folytatódjon!
- LM listából töröljük azokat, amelyek módosítása végrehajtható volt, hagyjuk benne azokat, amelyek hibás adatokat tartalmaztak. Írjon programot, mely input fájlból (fájlokból) felépíti a listákat (a fájl nincs rendezve árukód kód szerint!) majd előállítja a kért listákat, végül kiírja a tartalmukat. Műveletigény:  $O(n+m)$ , ahol  $n$  az L,  $m$  az LM lista hossza.

## Tervezés

### itemInf

<pre>+key: ℕ +measure: ℕ +next: itemInfo +iteminfo() {next=0}</pre>
---

### Items

<pre>-Status:enum { abnorm, norm } - _sx:Status - _dx: itemInfo - _x: fstream - read():ℤ -createList() -insertKey() {}</pre>
<pre>+Items(string filename) +~Items() +mergeList(Item *lm) +errors: enum { FILE_NAME_ERROR,EXISTING_KEY,OUT_OF_MEMORY}</pre>

h2e9aq

[h2e9aq@inf.elte.hu](mailto:h2e9aq@inf.elte.hu)

2. csoport

**createList()**

read()
insertKey()

**insertKey ()**

$pe=L, p=L->next$		
$p \neq 0 \wedge \_dx.key > p->key$		
$pe=p$		
$p=p->next$		
$p! \neq 0 \wedge \_dx.key == p->key$		
$EXISTING\_KEY$		
$curr=new$		
$curr->key=\_dx.key$		
$curr->measure=\_dx.measure$		
$pe->next=curr$		
$curr->next=p$		

**~Items ()**

$pe=L->next$		
$pe \neq 0$		
$p=pe->next$		
$delete\ pe$		
$L->next=p$		
$pe=p$		
$delete\ L$		

**mergeList()**

$p=L->next, \quad q=lm->L->next, \quad qe=lm->L$			
$q \neq 0$			
$p==0 \text{ // } p->$ $key>q->key$	$p->key=q->key$		
$qe=q$ $q=qe->next$	$p->measure<q->measure$		$p=p->next$
	$qe=q$	$p->measure-=q->measure$ $qe->next=q->next$ $q->next=0$	
	$q=qe->next, \quad p=p->next$		

## Teszt tervezés

1. Üres fájlra tesztelés(inp0.txt)  
**kiindulási adatok:** fájl üres  
**elvárt eredmény:** L->next=0
2. Rossz fájlnev kivétel elkapása(inp100.txt).  
**kiindulási adatok:** fájl nem létezik  
**elvárt eredmény:** error er=Items::FILE\_NAME\_ERROR
3. Beolvasandó fájlban többször szereplő kulcs, már létező kulcs kivétel elkapása(inp.txt).  
**kiindulási adatok:** inp.txt  
1 2  
3 4  
2 1  
0 2  
5 1  
10 2  
2 1  
**elvárt eredmény:** error er=Items:: EXISTING\_KEY
4. Összes érték beolvasásának ellenőrzése(mainList.txt).  
**kiindulási adatok:** mainList.txt (key, measure)  
1 2  
3 4  
2 1  
0 2  
5 1  
10 2  
**elvárt eredmény:** Listába bekerült értékek(key, measure):  
0 2  
1 2  
2 1  
3 4  
5 1  
10 2
5. Egy normál, hiba nélküli működést reprezentáló futtatás(l=mainList.txt, lm=soldList.txt).  
**kiindulási adatok:**  
L lista (key, measure):  
0 2  
1 2  
2 1  
3 4  
5 1  
10 2

h2e9aq

[h2e9aq@inf.elte.hu](mailto:h2e9aq@inf.elte.hu)

2. csoport

LM lista (key, measure):

1 2

2 1

3 3

**elvárt eredmény:**

6. L lista (key, measure):

0 2

1 0

2 0

3 1

5 1

10 2

LM lista üres.

7. LM listában az egyik kulcsnál több a kiviteli mennyiség, mint amennyi raktáron van (l=mainList.txt, lm=inp2.txt).

**kiindulási adatok:**

L lista (key, measure):

0 2

1 2

2 1

3 4

5 1

10 2

LM lista (key, measure):

2 1

3 4

100 2

**elvárt eredmény:**

L lista (key, measure):

0 2

1 2

2 0

3 1

5 1

10 2

LM lista (key, measure):

100 2

h2e9aq

[h2e9aq@inf.elte.hu](mailto:h2e9aq@inf.elte.hu)

2. csoport

9. Egy olyan kulcs szerepel az LM listában, amely nem létezik az L-ben. (l=mainList.txt, lm=inp3.txt).

**kiindulási adatok:**

L lista (key, measure):

0 2

1 2

2 1

3 4

5 1

10 2

LM lista (key, measure):

1 2

2 1

3 5

**elvárt eredmény:**

L lista (key, measure):

0 2

1 0

2 0

3 4

5 1

10 2

LM lista (key, measure):

3 5