# 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:

1. 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.
2. Ha olyan árura érkezik kivitel, amelyik nem szerepel a listában, szintén jelezzünk hibát.
3. Hibajelzés után az algoritmus folytatódjon!
4. 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**

|  |
| --- |
|  |
| +key: ℕ  +measure: ℕ  +next: itemInfo  +iteminfo() {next=0} |

**Items**

|  |
| --- |
| -Status:enum { abnorm, norm }  -\_sx:Status  -\_dx: itemInfo  - \_x: fstream  - read():𝕃  -createList()  -insertKey() {} |
| +Items(string filename)  +mergeList(Item \*lm)  +errors: enum { FILE\_NAME\_ERROR,EXISTING\_KEY,OUT\_OF\_MEMORY} |

**createList()**

|  |  |
| --- | --- |
| read() | |
|  | insertKey() |

**insertKey ()**

|  |  |  |
| --- | --- | --- |
| *pe=L, p=L->next* | | |
| *p ≠0 ⋀ \_dx.key>p->key* | | |
|  | *pe=p* | |
|  | *p=p->next* | |
|  | *p! ≠0 ⋀ \_dx.key==p->key* | |
|  | *EXISTING\_KEY* |  |
| *curr=new* | | |
| *curr->key=\_dx.key* | | |
| *curr->measure=\_dx.measure* | | |
| *pe->next=curr* | | |
| *curr->next=p* | | |

**mergeList()**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *p=L->next, q=lm->L->next, qe=lm->L* | | | | |
| q*≠*0 | | | | |
|  | *p==0 || 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, p=p->next* | |

## Teszt tervezés

1. Üres fájlra tesztelés(inp0.txt)  
   **kiindulási adatok:** fájl üres  
   **elvárt eredmény:** L->next=0
2. Rossz fájlnév 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

1. 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

1. 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

LM lista (key, measure):

1 2

2 1

3 3  
**elvárt eredmény:**

1. L lista (key, measure):

0 2

1 0

2 0

3 1

5 1

10 2

LM lista üres.

1. 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

1. 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