

Listák ...

*(Beadandó feladatok  
Programozásmódszertan 2. félévéhez,  
2005 tavasza)*

## Általános tudnivalók

- A feladatok lényege: megfelelő *típusok*, *adatszerkezetek* használata.
- A feladat szövegéből kiderül, hogy *milyen adatszerkezet* van a középpontban (pl. *láncoltan ábrázolt lista*). Ettől **nem szabad eltérni!**
- A feladatokban rendszerint előfordul valamilyen *felsorolási típus*, amelyet szintén, mint típust, meg kell valósítani *saját beolvasó* és *kiíró műveleteivel* együtt. Ugyanez vonatkozik a *dátum*, az *idő* (stb. „speciális”) típusokra is. Az *input-formátumra* vonatkozó elvárásokat a feladatok példái egyértelműen mintázzák.
- Cél a kijelölt *típusok* (minél inkább) *feladatfüggetlen megvalósítása*. Ez elsősorban az „elkülönített formába öntöttséget”, *különálló file-ba foglalását* jelenti<sup>1</sup>. Kapcsolat persze van a „fő” program és a típust megvalósító modul között: a modul paraméterei (befeled irányuló) és szolgáltatásai (kifelé irányuló).
- Mivel a program *kódja legalább kettő* (de inkább több) *file-ra* osztva alkot egy egészet, lényeges, hogy ezek együttese ne keveredjen más célú file-okkal, azaz a korábban *megkívánt file-struktúra létrehozása és figyelembevétele* most még inkább indokolt.
  - \ -- a gyökérben a futtatási környezet (EXE + adatfile-ok)
  - \FORRAS\ -- PAS programállomány + UNIT/INCLUDE forrásállományok
  - \DOKU\ -- DOC-állományok
- A program a szükséges *input-adatokat* elsősorban *file-okban várja* (bár ókor-ókor biztosítani kell a „kézi” paraméterezhetőséget is), ezért az *input-file-ok szerkezetére rótt elvárások* tiszteletben tartása alapvető *feltétele* a kipróbálásnak, *az érdemi értékelésnek*.
- A *teszteléshez „generált” file-ok* a *gyökérfiókban* legyenek, ott ahol az exe-állomány van. Nyilván ezekre történik utalás a dokumentáció teszteléssel foglalkozó részében. *Elvárás*, hogy a feladat szempontjából *lényeges esetekhez tartozzon* egy-egy *teszt-adatsor*. (Ha az exe mellett nincs adatállomány vagy csak 1-2 árválkodik, akkor ezt annak jeleként értékeljük, hogy a program nem igazán lett kipróbálva. L. az értékelési [útmutatót!](#))

---

<sup>1</sup> Ehhez segítségül felhasználható egy „[Cserél](#)” nevű program. ([ZIP](#))

## Feladatregiszter:

1. Labor-foglaltság vizsgálata <i>kétirányú, láncolt listával</i> .....	4
2. Cégvezetők elfoglaltságainak kezelése <i>kétirányú, láncolt listával</i> , programfajta figyelembe vételével .....	6
3. Szótár-vizsgálat – <i>láncolt listák</i> átszervezése .....	8
4. Régészeti relikviák vizsgálata – <i>láncolt listák</i> fajták szerinti szétválogatása .....	9
5. Borpince – <i>láncolt listák</i> szétválogatása .....	10
6. Labor-foglaltság vizsgálata <i>kétirányú, láncolt listával</i> , órafajták figyelembe vételével. ....	12
7. Üdülések nyilvántartása <i>kétirányú, láncolt listával</i> .....	14
8. Szótár-vizsgálat – <i>láncolt listák</i> átszervezése .....	16
9. Régészeti relikviák vizsgálata – <i>láncolt listák</i> szűrése .....	17
10. Borpince – <i>láncolt listák</i> összefuttatása .....	18
11. Labor-foglaltság vizsgálata <i>kétirányú, láncolt listával</i> .....	20
12. Ünnepek nyilvántartása <i>kétirányú, láncolt listákkal</i> .....	22
13. Szótár-vizsgálat – keresés <i>láncolt listákban</i> .....	24
14. Borpince – <i>láncolt listák</i> szétválogatása .....	26
15. Talált Tárgyak Osztálya – <i>láncolt listák</i> szétválogatása kategóriák szerint .....	27
16. Cégvezetők elfoglaltságainak kezelése <i>kétirányú, láncolt listával</i> , fontosság figyelembe vételével. ....	29
17. Szótár-vizsgálat – <i>láncolt listák</i> rendezése .....	31
18. Régészeti relikviák vizsgálata – <i>láncolt listák</i> értékliséje .....	32
19. Borpince – <i>láncolt listák</i> szétválogatása .....	33
20. Talált Tárgyak Osztálya – <i>láncolt listák</i> szétválogatása .....	34

## 1. Labor-foglaltság vizsgálata *kétirányú, láncolt listával*.

A laborok heti elfoglaltságát az alábbi szerkezetű elemeket tartalmazó *kétirányú, láncolt lista*ba foglalva tároljuk:

### Típus

```
TLaborFoglaltságok=Lista(TLaborFoglaltság)
TLaborFoglaltság=Rekord(Nap:TNap, Mettől,Meddig:TIdő,
                          Ki:TSzak),
TNap=(Hétfő,Kedd,Szerda,Csütörtök,Péntek,Szombat,
      Vasárnap) [az előadáson taglalt felsorolástípus]
TSzak=(Info, Prog, LC, Met, Fiz) [ez is felsorolástípus]
TIdő=Rekord(óra,perc:Egész)
```

Laboronként *külön-külön listát* vezet a foglaltságról. A listákban TNap, s azon belül TIdő szerint rendezve vannak a foglaltsági adatok. **Módosítsa** a listákat új, file-ból jövő bejegyzésekkel. (Kezdetben üresek a listák.) Azaz ha eddig még szabad időpontra vonatkozó igény jött, illessze be; ha –részben vagy egészben– foglalt a megjelölt időintervallum, szüntesse meg az érintett összes foglaltságot, majd helyettesítse az újjal, s a „*kitúrtak*” egy másik *listába* fűzze! Kérésre **jelentse meg** a heti elfoglaltságát egy adott labornak, és az addig „*kitúrtak*” adatait (mely időpontban, mely laborból, mikori dátumú igénylés alapján).

### Típus

```
TKitúrtak=Lista(TKitúrt)
TKitúrt=Rekord(Labor:TLabor,
                érkiIdő:TDátum, Szak:TSzak, Mettől:TIdő)
```

### A bemenet:

- o teremigénylések sorozata szöveges file-ban – az igénylés-file egy eleme a file 6 egymásutáni sorában (igénylési idő szerint növekvő sorrendben):

```
igénylésIdő:TDátum [amikor az igénylést beadták]
labor:TLabor
nap:TNap
mettől:TIdő
meddig:TIdő
szak:TSzak
```

Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság. Az eddig nem definiált típusokról a következőket kell tudni:

```
TDátum=Rekord(év,hó,nap:Egész)
TLabor=(PC1,PC2,PC3,PC5,PC5,MM,Grafika,MS) [ez is felsorolástípus]
```

A TDátum típus input-szintaxisa: Egész+SZÓKÖZ+Egész+SZÓKÖZ+Egész.

### *Példa – két igénylés az input-file-ból:*

```
2001 3 7
PC1
HÉTFŐ
7 0
9 45
INFO
```

Egy 2001. március 7-diki igénylés...

```
2001 3 17
PC2
PÉNTEK
12 15
16 30
PROG
```

... és egy másik ...

- Klaviatúra input: egy dátum és egy laborazonosító, amikor, és amelyik labor teremfoglaltsági állapotot kéri kiírni.

***Egy példa – klaviatúra-input:***

```
2001 3 7
MM
```

Egy 2001. március 7. éjféli állapotkérés...

## 2. Cégvezetők elfoglaltságainak kezelése *kétirányú, láncolt listával*, programfajta figyelembe vételével.

Egy cég néhány vezetőjének közeljövőbeli (két adott dátum közötti) elfoglaltságait az alábbi szerkezetű elemeket tartalmazó *kétirányú, láncolt listába* foglalva tároljuk:

```
Típus
TElfooglaltságok=Lista(TElfooglaltság)
TElfooglaltság=Rekord(
    érkiDő:TDátum [az igény érkezési ideje],
    Ki:TNév, Nap:TDátum, Mettől:TIdő, Perc:Egész,
    Ok:TProgram),
TProgram=(Megbeszélés,Előadás,Ebéd,Party)
TDátum=Rekord(év,hó,nap:Egész)
TIdő=Rekord(óra,perc:Egész)
```

Vezetőnként *külön-külön listát* kell vezetni a foglaltságról. A listában *érkiDő:TDátum, Nap:TDátum*, s azon belül *Mettől:TIdő* szerint rendezve vannak a foglaltsági adatok. **Építse föl** a listákat a file-ból jövő program-kérések alapján. Ha eddig még szabad időpontra vonatkozó programajánlat jött, illesse be a megfelelő –időrendbeli– helyre. Ha azonban –részben vagy egészben– foglalt a megjelölt időintervallum, akkor „fontosságuk” szerint szüntesse meg az érintett összes foglaltságot, és helyettesítse az újjal, vagy utasítsa vissza az igényt. A fontossági sorrend a legfontosabbal kezdve: Megbeszélés..Party. Az esetleg *visszautasítottakat* (vagy „*kitúrtakat*”) egy másik *listába* fűzze! Ha olyan kérés érkezik, amely megérkezésénél korábbi időpontra vonatkozik, akkor azt fűzze a *rákérdezendők listájába*, Nap:TDátum, azaz a kért időpont szerint rendezetten! **Írja ki** az elfoglaltságát minden vezetőnek, és a visszautasított elfoglaltság-igénylések adatait, valamint a *rákérdezendők listáját!*

```
Típus
TVisszautasítottak=Lista(TVisszautasított)
TVisszautasított=Rekord(
    Név:TNév, érkiDő:TDátum,
    Nap:TDátum, Mettől:TIdő, Perc:Egész,
    Ok:TProgram,
    Miatt:TProgram [amilyen program miatt lett
                    visszautasítva])
TRákérdezendők=Lista(TRákérdezendő)
TRákérdezendő=Rekord(Név:TNév,
    érkiDő:TDátum, [az érkezés dátuma]
    Nap:TDátum [a program dátuma])
```

### A bemenet:

- o elfoglaltsági igénylések sorozata szöveges file-ban igénylési idő szerint rendezetten érkeznek – az igénylés-file egy eleme a file 7 egymásutáni sorában:

```
érkiDő:TDátum [amikor az igénylés befutott]
vezető:TvezetőNév [akire vonatkozik az igénylés]
partner:TNév [az igénylő]
nap:TDátum [amikorra vonatkozik: tetszőleges!!!]
mettől:TIdő
perc:Egész [mettől hány percet kér]
ok:TProgram [milyen jellegű az elfoglaltság]
```

Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni, de feltételezhető a hibátlanság. Az eddig nem definiált típusról a következőket kell tudni:

**Típus**

```
TVezetőNév=('Jó Isten','Luci Feri','Angyal Gábrriel')
           [ez is felsorolástípus,
           lehet persze más neveket is ide tenni]
```

**Példa – két igénylés az input-file-ból:**

```
1999 12 27
Luci Feri
Belze Bub
1999 12 31
23 0
59
BESZÉLGETÉS
```

Egy 2001. március 7-diki igénylés...

```
2000 1 7
Jó Isten
Assisi Szent Ferenc
2000 1 1
12 15
1
ELŐADÁS
```

... és egy másik ...

(-: *Nincs tévedés: egy korábbi dátumra vonatkozó kérésről van szó. Lassú volt a postás. :-)*)

### 3. Szótár-vizsgálat – *láncolt listák* átszervezése

A kétnyelvű szótárak –leegyszerűsítve– tekinthetők szópárok listájának, ahol az első szó a tárgynyelv (pl. angol) szava, a második pedig a célnyelvé (pl. magyar). Természetesen előfordulhat, hogy egy szónak több párja is van. Ez azt jelenti, hogy ugyanaz a szó több párosításban is szerepel. Feltehető, hogy a szó (mind a tárgy, mind a célnyelvi „oldalon”) egyetlen szóból áll, hossza nem haladja meg a 20 karaktert. Gyakran előfordulnak azonos alakú, de különböző szófajú szavak (pl. ír – ige, ír – jelző), ezek megkülönböztethetősége érdekében a szópár mellett közöljük a szófajt is.

Beolvasva a szópárokat, amelyeket az input két egymást követő sora tartalmaz, felépít egy *láncolt listát* (a „bemeneti szótárat”). Majd szervezze át *dinamikus* a szótárat („szócikk-tár”) úgy, hogy egy szócikkbe sorolja azokat a célszavakat, amelyeknek azonos a tárgyszava. (Például, ha előfordul a szótárban az (ír, writes) és az (ír, Irish) szópár, akkor ők együtt alkotnak egy szócikket: ír → writes, Irish. Ez tehát egy célszó-lista lesz.) Írja ki a szócikkbe szervezett szótárat a képernyőre lapozottan, *táblázatosan!*

A *dinamikus* határozóval arra utaltam, hogy amint egy célszóval végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

#### Típus

```
TBeSzótár=Lista (TSzópár)
TBeSzópár=Rekord (tSzó, cSzó: Szöveg, szófaj: TSzófaj)
TSzófaj= (Ige, Főnév, Határozó, Jelző) [felsorolástípus]
TSzócikk-tár=Rekord (tSzó: Szöveg, cSzavak: TCélszavak)
TCélszavak=Lista (TCélszó)
TCélszó=Rekord (cSzó: Szöveg, szófaj: TSzófaj)
```

#### A bemenet:

- o a szópárokat tartalmazó text-file (\*.DIC):

```
tárgyszó + szóköz + célszó: Szöveg
szófaj: TSzófaj
...
↵ Üres sor jelzi a feldolgozás végét.
```

Természetesen a szófaj beolvasása a TSzófaj típus 'Be:' műveletével végzendő el!

#### A kimenet:

- o Szócikk-tár:

```
tárgyszó -> célszó1 (szófaj1)
           célszó2 (szófaj2)
           ...
...
```

Ügyeljen a képernyő fenti „strukturáltságára”! Természetesen a szófaj kiírása a TSzófaj típus 'Ki:' műveletével végzendő el!



#### 4. Régészeti relikviák vizsgálata – *láncolt listák* fajták szerinti szétválogatása

Egy ásatáson gyűjtött tárgyakról feljegyzik: a fajtáját (korsó, tányér stb.), épségét (százalék), anyagát (ami bizonyos alapanyagok halmazát jelenti) és becsült korát.

Olvassa be egy *láncolt listába*, majd fajtánként külön-külön szétválogatja *dinamikusan*. E listákat kiírja a képernyőre *rendezetten* és lapozottan!

A *dinamikusan* határozóval arra utaltam, hogy amint egy tárggyal végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

A fajta-listák rendezését épség szerint csökkenőleg végzi el.

##### Típus

```
TrTárgyak=Lista(TrTárgy)
TrTárgy=Rekord(fajta:TFajta,épség:Egész [százalék],
                anyag:TAnyag, kor:Egész)
TFajta=(Korsó,Tányér,Kés,Kard,Tőr,Pohár,Egyéb)
TAnyag=Halmaz(TAlapanyag)
TAlapanyag=(Bronz,Vas,Réz,Agyag,Ón,Szövet,Más)
```

A fenti felsorolás és halmaz típusokat önálló modulokban kell meg valósítani. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

##### A bemenet:

- o A tárgyakat tartalmazó text-file (\*.REL):

```
Tárgyak száma
Fajta1
Épség1
Anyag11 + vessző + Anyag12 + ...
Kor1
Fajta2
Épség2
Anyag21 + vessző + Anyag22 + ...
Kor2
...
```

##### A kimenet:

- o A tárgyak fajták szerinti listája:

```
Fajta1
* Épség11 , Kor11
  Anyag111 , Anyag112 ...
* Épség12 , Kor12
  Anyag121 , Anyag122 ...
...
Fajta2
* Épség21 , Kor21
  Anyag211 , Anyag212 ...
...
```

Ügyeljen az elválasztójelekre (csillag, vessző)!

## 5. Borpince – *láncolt listák* szétválogatása

Egy borpincészet a borairól feljegyzi a következő adatokat: fajta (vörös, fehér, rozé), megnevezése, borvidék (Alföldi, Badacsonyi, Egri, Mátrai, Móri, Soproni, Tokaji, Villányi), évjárat, cukortartalom, szesztartalom, kiszérelés (hordó, üveg7, üveg5).

Olvassa be egy *láncolt listába*, majd válogassa szét a szerint, hogy mely évtizedbe sorolandó az adott bor. A szétválogatás a nagyszámú listára való tekintettel *dinamikus*an kell elvégeznie.

E listákat kiírja a képernyőre fajták szerint *rendeztetve*, lapozottan!

A *dinamikus*an határozóval arra utaltam, hogy amint egy borral végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

Meghatározza, hogy az egyes évtizedekbe hány bor tartozik, továbbá az átlagos cukorfokot (szintén évtizedenként)!

### Típus

```
TPincészet=Lista(TBor)
TBor=Rekord(fajta:TFajta,név:Szöveg,vidék:TVIDék,
             évjárat:Egész,cukor,szesz:Valós [tömegszázalék],
             kiszérelés:TKiszérelés)
TFajta=(vörös,fehér,rozé)
TVIDék=(Alföldi,Badacsonyi,Egri,Mátrai,Móri,Soproni,
        Tokaji,Villányi)
TKiszérelés=(hordó,üveg7,üveg5)
```

A fenti felsorolás típusokat önálló modulokban meg kell valósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanúság.

### A bemenet:

- o A borokat tartalmazó text-file (\*.BOR):

```
1. sor: Borok száma
2. sor: Fajta1 + Megnevezés1 + Vidék1 + Évjárat1 + vessző +
        Cukor1 + vessző + Szesz1 + Kiszérelés1
3. sor: Fajta2 + Megnevezés2 + Vidék2 + Évjárat2 + vessző +
        Cukor2 + vessző + Szesz2 + Kiszérelés2
...
```

### A kimenet:

- o A borok dekád-listája:

```
Évtized-kezdet1 - Évtized-vég1
* Megnevezés11 , Fajta11
  Vidék11 , Évjárat11 , Cukorfok11 , Szeszfok11
* Megnevezés12 , Fajta12
  Vidék12 , Évjárat12 , Cukorfok12 , Szeszfok12
...
Évtized-kezdet2 - Évtized-vég2
* Megnevezés21 , Fajta21
  Vidék21 , Évjárat21 , Cukorfok21 , Szeszfok21
* Megnevezés22 , Fajta22
  Vidék22 , Évjárat22 , Cukorfok22 , Szeszfok22
...
```

Ügyeljen az elválasztójelekre (csillag, vessző)!

- A borok fajták szerinti jellemzése (legidősb bor, átlag-cukorfok):

$\text{Évtized-kezdet}_1 - \text{Évtized-vég}_1 * \text{borszám}_1 * \text{átlagcukor}_1$
$\text{Évtized-kezdet}_2 - \text{Évtized-vég}_2 * \text{borszám}_2 * \text{átlagcukor}_2$
...

Ügyeljen az elválasztójelekre (csillag)!

## 6. Labor-foglaltság vizsgálata *kétirányú, láncolt listával*, órafajták figyelembe vételével.

A laborok heti elfoglaltságát az alábbi szerkezetű elemeket tartalmazó *kétirányú, láncolt lista*ba foglalva tároljuk:

### Típus

```
TLaborFoglaltságok=Lista(TLaborFoglaltság)
TLaborFoglaltság=Rekord(Nap:TNap, Mettől,Meddig:TIdő,
                          Ki:TSzak, Fajta:Tfajta),
TNap=(Hétfő,Kedd,Szerda,Csütörtök,Péntek,Szombat,
      Vasárnap) [az előadáson taglalt felsorolástípus]
TSzak=(Info,Prog,LC,Met,Fiz) [ez is felsorolástípus]
TFajta=(Egyszeri,Reguláris,Modul,Speci)
TIdő=Rekord(óra,perc:Egész)
```

Laboronként *külön-külön listát* vezet a foglaltságról. A listában TNap, s azon belül TIdő szerint rendezve vannak a foglaltsági adatok. **Módosítsa** a listákat új, file-ból jövő bejegyzésekkel. (Kezdetben üresek a listák.) Azaz ha eddig még szabad időpontra vonatkozó igény jött, illessze be; ha –részben vagy egészben– foglalt a megjelölt időintervallum, akkor „fontossága” szerint szüntesse meg az érintett összes foglaltságot, és helyettesítse az újjal, vagy utasítsa vissza az igényt. A fontossági sorrend a legfontosabbal kezdve: Egyszeri.. Speci. Az esetleg *visszautasítottakat* egy másik *listába* fűzze! Kérésre **jelentse meg** a heti elfoglaltságát egy adott labornak, és az addig arra a laborra vonatkozó, visszautasított igénylések adatait (igénylési adatok és annak az órának az adatai –elegendő egynek, ha több is volt ilyen–, amely megakadályozta a foglalást).

### Típus

```
TVisszautasítottak=Lista(TVisszautasított)
TVisszautasított=Rekord(
    Labor:TLabor,
    érkJdő:TDátum, VSzak:TSzak, VFajta:Tfajta,
    Mettől,Meddig:TIdő,
    Szak:TSzak [emiatt lett visszautasítva],
    Fajta:TFajta)
```

### A bemenet:

- o teremigénylések sorozata szöveges file-ban – az igénylés-file egy eleme a file 7 egymástutáni sorában (igénylési idő szerint növekvő sorrendben):

```
igénylésIdő:TDátum [amikor az igénylést beadták]
labor:TLabor
nap:TNap
mettől:TIdő
meddig:TIdő
szak:TSzak
fajta:TFajta
```

Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság. Az eddig nem definiált típusokról a következőket kell tudni:

```
TDátum=Rekord(év,hó,nap:Egész)
TLabor=(PC1,PC2,PC3,PC5,PC5,MM,Grafika,MS) [ez is
      felsorolástípus]
```

A TDátum típus input-szintaxisa: Egész+SZÓKÖZ+Egész+SZÓKÖZ+Egész.

**Példa – két igénylés az input-file-ból:**

```
2001 3 7
PC1
HÉTFŐ
7 0
9 45
INFO
REGULÁRIS
```

Egy 2001. március 7-diki igénylés...

```
2001 3 17
PC2
PÉNTEK
12 15
16 30
PROG
SPECI
```

... és egy másik ...

- Klaviatúra input: egy dátum és egy laborazonosító, amikor, és amelyik labor teremfoglaltsági állapotot és az addig visszautasítottak listáját kéri kiírni.

**Egy példa – klaviatúra-input:**

```
2001 3 7
PC2
```

Egy 2001. március 7. éjféli állapotkérés...



2001 4 1 PéNZ Elek 2001 7 13 5 2 KASTÉLY
2001 4 1 Hamari Húgó 2001 3 31 14 4 PUTRI

Pénzes hp. a Kastélyban nyaralna július 13.-  
ától 5 napig. Mindez egy április 1.-ei meg-  
rendelésükből derül ki.

Hamariék második megrendelésük –való-  
színűsíthetően postai okok miatt– csak ápri-  
lis elsején érkezett meg, pedig március 31.-  
től 14 napon át tavaszoltak *volna* a Putri-  
ban.

A *visszautasított* megrendeléseket egy külön *listában* halmozzuk föl:

<p><b>Típus</b> TVisszamondások=<b>Lista</b> (TVisszamondás) TVisszamondás=<b>Rekord</b> (Név:TNév, Mettől:TDátum, Azonosító:TÜdülő, Ok: (JobbAjánlat,Késett) )</p>
---

Kérésre **jelenítse meg** egy adott időpontban érvényes szoba-foglaltsági adatokat, és az addig visszautasított igénylések adatait.

**Egy példa – klaviatúra-input:**

2001 4 2
----------

A 2001. április 2. éjféli állapotkérés...

## 8. Szótár-vizsgálat – *láncolt listák* átszervezése

A kétnyelvű szótárak –leegyszerűsítve– tekinthetők szópárok listájának, ahol az első szó a tárgynyelv (pl. angol) szava, a második pedig a célnyelvé (pl. magyar). Természetesen előfordulhat, hogy egy szónak több párja is van. Ez azt jelenti, hogy ugyanaz a szó több párosításban is szerepel. Feltehető, hogy a szó (mind a tárgy, mind a célnyelvi „oldalon”) egyetlen szóból áll, hossza nem haladja meg a 20 karaktert. Gyakran előfordulnak azonos alakú, de különböző szófajú szavak (pl. ír – ige, ír – jelző), ezek megkülönböztethetősége érdekében a szópár mellett közöljük a szófajt is.

Beolvasva a szópárokat, amelyeket az input két egymást követő sora tartalmaz, felépít egy *láncolt listát* (a „bemeneti szótárat”), majd szervezze át *dinamikus*an a szótárat („szócikk-tár”) úgy, hogy egy szócikkbe sorolja azokat a szavakat, amelyek „összertartoznak”. (Például, ha előfordul a szótárban az (ír, writes) és az (ír, Irish) szópár, akkor ők összertartoznak, együtt alkotnak egy szócikket: ír → writes, Irish. Ez tehát egy olyan szó-lista lesz, amelynek első szava a tárgyszó, a többiek a hozzátartozó célszavak.) Írja ki a szócikkbe szervezett szótárat a képernyőre lapozottan, *táblázatosan!*

A *dinamikus*an határozóval arra utaltam, hogy amint egy célszóval végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

### Típus

```
TBeSzótár=Lista(TSzópár)
TBeSzópár=Rekord(tSzó,cSzó:Szöveg, szófaj:TSzófaj)
TSzófaj=(Ige,Főnév,Határozó,Jelző,NemDef) [felsorolástípus]
TSzócikktár=Lista(TTárgyVagyCélszó)
      [Típusinvariáns: l:TSzócikktár | Hossz(l)>1
      az első elem a tárgyszó]
TTárgyVagyCélszó=Rekord(szó:Szöveg, szófaj:TSzófaj)
      [tárgyszó esetén a szófaj=NemDef]
```

### A bemenet:

- o a szópárokat tartalmazó text-file (\*.DIC):

```
tárgyszó + szóköz + célszó:Szöveg
szófaj:TSzófaj
...
↓ Üres sor jelzi a feldolgozás végét.
```

Természetesen a szófaj beolvasása a TSzófaj típus 'Be:' műveletével végzendő el!

### A kimenet:

- o Szócikk-tár:

```
tárgyszó -> célszó1 (szófaj1)
           célszó2 (szófaj2)
           ...
           ...
```

Ügyeljen a képernyő fenti „strukturáltságára”! Természetesen a szófaj kiírása a TSzófaj típus 'Ki:' műveletével végzendő el!



## 9. Régészeti relikviák vizsgálata – *láncolt listák* szűrése

Egy ásatáson gyűjtött tárgyakról feljegyzik: a fajtáját (korsó, tányér stb.), épségét (százalék), anyagát (ami bizonyos alapanyagok halmazát jelenti) és becsült korát.

Olvassa be egy *láncolt listába*, majd fajtánként külön-külön szétválogatja, *dinamikusan*. E listákat *rendezi kor szerint*, majd kiírja a képernyőre lapozottan!

A *dinamikusan* határozóval arra utaltam, hogy amint egy tárggyal végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

Meghatározza, hogy az egyes fajtáknak mennyi az átlagkora!

### Típus

```
TrTárgyak=Lista(TrTárgy)
TrTárgy=Rekord(fajta:TFajta,épség:Egész [százalék],
                anyag:TAnyag, kor:Egész)
TFajta=(Korsó,Tányér,Kés,Kard,Tőr,Pohár,Egyéb)
TAnyag=Halmaz(TAlapanyag)
TAlapanyag=(Bronz,Vas,Réz,Agyag,Ón,Szövet,Más)
```

A fenti felsorolás és halmaz típusokat önálló modulokban kell megvalósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

### A bemenet:

- o A tárgyakat tartalmazó text-file (\*.REL):

```
Tárgyakszáma
Fajta1
Épség1
Anyag11 + vessző + Anyag12 + ...
Kor1
Fajta2
Épség2
Anyag21 + vessző + Anyag22 + ...
Kor2
...
```

### A kimenet:

- o A tárgyak fajták szerinti listája (és átlag kora):

```
Fajta1 # átlagkor1
* Épség11 , Kor11
  Anyag111 , Anyag112 +
* Épség12 , Kor12
  Anyag121 , Anyag122 +
...
Fajta2 # átlagkor1
* Épség21 , Kor21
  Anyag211 , Anyag212 +
...
```

Ügyeljen az elválasztójelekre (kettős kereszt, csillag, vessző)!

## 10. Borpince – *láncolt listák* összefuttatása

Borpincészetek a boraikról feljegyzik a következő adatokat: fajta (vörös, fehér, rozé), megnevezése, borvidék (Alföldi, Badacsonyi, Egri, Mátrai, Móri, Soproni, Tokaji, Villányi), évjárat, cukortartalom, szesztartalom, kiszereles (hordó, üveg7, üveg5).

Olvassa be *láncolt listák*ba a paraméterként megadott pincészet nyilvántartását, majd rendezze ezeket kiszereles szerint! E listákat kiírja a képernyőre lapozottan! Ezek után készítse egy olyan listát, amelyben az előbbi listák elemei kerülnek! Az összefuttatás több listára vonatkozik, ezért a részfeladatot *dinamikusan* kell elvégeznie.

A *dinamikusan* határozóval arra utaltam, hogy amint egy borral végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

Meghatározza, hogy hány tétel tartozik az egyes kiszerelesekbe!

### Típus

```
TPincészet=Lista (TBor)
TBor=Rekord (fajta:TFajta,név:Szöveg,vidék:TVIDék,
              évjárat:Egész,cukor,sesz:Valós [tömegszázalék],
              kiszereles:TKiszereles)
TFajta=(vörös,fehér,rozé)
TVIDék=(Alföldi,Badacsonyi,Egri,Mátrai,Móri,Soproni,
        Tokaji,Villányi)
TKiszereles=(hordó,üveg7,üveg5)
```

A fenti felsorolás típusokat önálló modulokban meg kell valósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanúság.

### A bemenet:

- o A pincészet nyilvántartását tartalmazó file-ok azonosítása (klaviatúra-input):

```
File-név1
File-név2
...
↓ Üres sor jelzi a feldolgozás végét.
```

- o Egy pincészet nyilvántartását tartalmazó text-file (\*.BOR):

```
1. sor: Borok száma + szóköz + pincészet neve
2. sor: Fajta1 + Megnevezés1 + Vidék1 + Évjárat1 + vessző +
        Cukor1 + vessző + Szesz1 + Kiszereles1
3. sor: Fajta2 + Megnevezés2 + Vidék2 + Évjárat2 + vessző +
        Cukor2 + vessző + Szesz2 + Kiszereles2
...
```

### A kimenet:

- o Az összes bor dekád-listája:

```
Évtized-kezdés1 - Évtized-vég1
* Megnevezés11 , Fajta11
  Vidék11 , Évjárat11 , Cukorfok11 , Szeszfok11
* Megnevezés12 , Fajta12
  Vidék12 , Évjárat12 , Cukorfok12 , Szeszfok12
...
```

Évtized-kezd<sub>2</sub> - Évtized-vég<sub>2</sub>  
\* Megnevezés<sub>21</sub> , Fajta<sub>21</sub>  
  Vidék<sub>21</sub> , Évjárat<sub>21</sub> , Cukorfok<sub>21</sub> , Szeszfok<sub>21</sub>  
\* Megnevezés<sub>22</sub> , Fajta<sub>22</sub>  
  Vidék<sub>22</sub> , Évjárat<sub>22</sub> , Cukorfok<sub>22</sub> , Szeszfok<sub>22</sub>  
...

Ügyeljen az elválasztójelekre (csillag, vessző)!

- o A borok kiszerelés szerinti jellemzése (darabszám):

hordó \* borszám<sub>hordó</sub>  
üveg7 \* borszám<sub>üveg7</sub>  
üveg5 \* borszám<sub>üveg5</sub>

Ügyeljen az elválasztójelekre (csillag)!

**11.** Labor-foglaltság vizsgálata *kétirányú, láncolt listával*.

A laborok heti elfoglaltságát az alábbi szerkezetű elemeket tartalmazó *kétirányú, láncolt lista*ba foglalva tároljuk:

**Típus**

```
TLaborFoglaltságok=Tömb (TNap:TLaborFoglaltság)
TLaborFoglaltság=Lista (TNapirend) ,
TNapirend=Rekord (Mettől,Meddig:TIdő, Ki:TSzak) ,
TNap=(Hétfő,Kedd,Szerda,Csütörtök,Péntek,Szombat,
      Vasárnap) [az előadáson taglalt felsorolástípus]
TSzak=(Info,Prog,LC,Met,Fiz) [ez is felsorolástípus]
TIdő=Rekord (óra,perc:Egész)
```

Laboronként *külön-külön listát* vezet a foglaltságról. A listában TNap, s azon belül TIdő szerint rendezve vannak a foglaltsági adatok. **Építse föl** a listákat a file-ból érkező adatok alapján. Ha egy foglalási igény –részben vagy egészben– ütközik a már fölvettek valamelyikével, akkor azt egy másik „*teljesíthetetlen igények*” *listába* gyűjti!

Kérésre **jelenítse meg** egy adott időpontban érvényes heti foglaltsági állapotát a laboroknak, naponkénti kitöltöttségi arányokat (%-ban), és az addig visszautasított igénylések adatait (igénylési adatok és annak az órának az adatai –ha több is van, elegendő közülük egy–, amely megakadályozta a foglalást).

**Típus**

```
TNemTeljesíthetők=Lista (TNemTeljesíthető)
TNemTeljesíthető=Rekord (
      Labor:TLabor, érkJdő:TDátum,
      NTSzak:TSzak, Mettől,Meddig:TIdő,
      Szak:TSzak [emiatt lett visszautasítva])
```

**A bemenet:**

- o teremigénylések sorozata szöveges file-ban – az igénylés-file egy eleme a file 6 egymásutáni sorában (igénylési idő szerint növekvő sorrendben):

```
igénylésIdő:TDátum [amikor az igénylést beadták]
labor:TLabor
nap:TNap
mettől:TIdő
meddig:TIdő
szak:TSzak
```

Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság. Az eddig nem definiált típusokról a következőket kell tudni:

```
TDátum=Rekord (év, hó, nap:Egész)
TLabor=(PC1,PC2,PC3,PC5,PC5,MM,Grafika,MS) [ez is
      felsorolástípus]
```

A TDátum típus input-szintaxisa: Egész+SZÓKÖZ+Egész+SZÓKÖZ+Egész.

**Példa – két igénylés az input-file-ból:**

```
2001 3 7
PC1
HÉTFŐ
7 0
9 45
INFO
```

Egy 2001. március 7-diki igénylés...

```
2001 3 17
PC2
PÉNTEK
12 15
16 30
PROG
```

... és egy másik ...

- Klaviatúra input: egy dátum és egy laborazonosító, amikor, és amelyik labor teremfoglaltsági állapotot és az addig visszautasítottak listáját kéri kiírni.

***Egy példa – klaviatúra-input:***

```
2001 3 7
MM
```

Egy 2001. március 7. éjféle állapotkérés...

## 12. Ünnepek nyilvántartása kétirányú, láncolt listákkal.

Nemzeti és családi ünnepeink dátumait egy *kétirányú, láncolt, ciklikus listában* tároljuk, időrendi sorrendben.

### Típus

```
TÜnnepek=Lista(TÜnnap)
TÜnnap=Rekord(Mikor:TDátum, Kör:TKör, Fajta:TFajta,
                Megjegyzés:Szöveg [pontosítás])
    [Típusinvariáns: Mikor:TDátum-szerint rendezve]
TDátum=Rekord(év,hó,nap:Egész)
TKör=(Családi,Baráti,Nemzeti,Egyéb)
TFajta=(Születésnap,Névnap,Szilveszter,Húsvét...).
```

Építsük föl a listát text-file-ből olvasott adatok alapján, amely össze-vissza tartalmazza az ünnepeket. A bemeneti file alábbi szerkezetű:

```
mikor:TDátum      [az ünnep időpontja]
kör:TKör          [milyen körben tekintendő ünnepnek]
fajta:TFajta     [ünnep megnevezése]
megjegyzés:Szöveg [további pontosítás: Anya (szülinapja)]
```

**Megjegyzés:**  
Érdeemes továbbiakkal kibővíteni ezt a felsorolást.

### Példa – ünnep file (SZP.UNN):

2000 4 24 EGYÉB HÚSVÉT	Április 24. = Húsvét., egyéb (értsd: „internacionális”) körben ünnepelt ünnep. Megjegyzés nincs, helyette üres a sor!
2000 1 1 EGYÉB SZILVESZTER	Január 1. = Szilveszter, egyéb (értsd: „internacionális”) körben ünnepelt ünnep. Megjegyzés nincs, helyette üres a sor!
2000 12 25 CSALÁDI SZÜLETÉSNAP Anya	December 25. = Anya születésnapja, családi körben ünnep.
2000 3 15 NEMZETI ÉVFORDULÓ 48-AS FORRADALOM	Március 15. = a ’48-as forradalom’ évfordulója, nemzeti ünnep.
2000 12 25 EGYÉB KARÁCSONY	December 25. = Karácsony, „több, mint” nemzeti ünnep. Megjegyzés nincs, helyette üres a sor!

Természetesen a fent szereplő adatokat a megfelelő típus ’Be:’/’Ki:’ műveleteivel kell beolvasni/kiírni, de feltételezhető a hibátlanság.

Legyen mód arra, hogy egy **adott dátumhoz kilistázzuk** azon **ünnepeket**, amelyek a dátum **8-napos környezetébe** esik (előre és vissza)! A dátumot klaviatúráról (akár ismétlődően) kéri a program:

### Egy példa – klaviatúra-input:

```
2000 3 22
```

2000. március 22. „környéki” ünneplista-kérés

Tegyük lehetővé az alábbi **karbantartó műveletek végrehajtását**:

1. adott ünnep *törlése* (azon ünnepekre vonatkozik, amelyekre teljesül a megadott minta),
2. új ünnep *fölvitele* (ha ilyen megjegyzésű volt már, akkor visszakérdezni, hogy cserélni kell, vagy újként beilleszteni),
3. bizonyos ünnepek *kilistázása* (azon ünnepekre vonatkozik, amelyekre teljesül a megadott minta)
4. az évszám *aktualizálása* (azaz minden évszám a megadottal helyettesítése)

Ez utóbbi funkciók végrehajtását egy karbantartó file vezérli, amely tartalmazza, hogy milyen funkciót kell végrehajtani, s –a 4. kivételével– egy ünnepminta, amely egy „komplett” ünnepmegadás a fentebb tisztázott szintaxissal.

**Példa – karbantartó file (SZP\_MOD.UNN):**

AKTUALIZÁLÁS 2002	2002-re áttérés
LISTÁZÁS * * HÚSVÉT *	Listát kérünk azokról az ünnepekről, amelyek fajtája: „HÚSVÉT”, a többi mezője tetszőleges.
TÖRLÉS 2002 4 24 * HÚSVÉT *	Töröljük az „egykori” (2000.-beli) Húsvétot.
ÚJ 2002 4 1 EGYÉB HÚSVÉT	Új ünnepként regisztráljuk a Húsvétot.

A karbantartó file nevét –természetesen– klaviatúráról kéri be (feltéve, hogy a felhasználó igényli).

### 13. Szótár-vizsgálat – keresés *láncolt listák*ban

A kétnyelvű szótárak –leegyszerűsítve– tekinthetők szópárok listájának, ahol az első szó a tárgynyelv (pl. angol) szava, a második pedig a célnyelvé (pl. magyar). Természetesen előfordulhat, hogy egy szónak több párja is van. Ez azt jelenti, hogy ugyanaz a szó több párosításban is szerepel. Feltehető, hogy a szó (mind a tárgy, mind a célnyelvi „oldal”) egyetlen szóból áll, hossza nem haladja meg a 20 karaktert. Gyakran előfordulnak azonos alakú, de különböző szófajú szavak (pl. ír – ige, ír – jelző), ezek megkülönböztethetősége érdekében a szópár mellett közöljük a szófajt is.

Beolvassa a szópárokat, amelyeket amelyek az input két egymást követő sora tartalmaz, felépít egy *láncolt listát* (szótárat), majd rendezzi a tárgynyelvi szó szerint növekvően. Írja ki a rendezett szótárat a képernyőre lapozottan, *táblázatosan!* Majd keressen ki a szótárból néhány, paraméterként adott szót, amelyek hovatartozása eldönthető a szótár alapján: vagy tárgyszóként vagy célszóként található meg. Ezekhez egy-egy *láncolt listát* generál, amely a keresett szót, és „fordításait” tartalmazza. A keresésnél ki kell használnia, hogy a lista, amiben keres, az egyik irányban rendezett.

#### Típus

```
TBeSzótár=Lista(TSzópár)
TBeSzópár=Rekord(tSzó,cSzó:Szöveg, szófaj:TSzófaj)
TSzófaj=(Ige,Főnév,Határozó,Jelző,NemDef) [felsorolástípus]
TSzófordítások=Lista(TSzófordítás)
TSzófordítás=Lista(TTárgyVagyCélszó)
                [Típusinvariáns: 1:TSzófordítás | Hossz(1)>1
                az első elem a tárgyszó]
TTárgyVagyCélszó=Rekord(szó:Szöveg, szófaj:TSzófaj)
                [tárgyszó esetén a szófaj=NemDef]
```

#### A bemenet:

- o a szópárokat tartalmazó text-file (\*.DIC):

```
tárgyszó + szóköz + célszó:Szöveg
szófaj:TSzófaj
...
↵ Üres sor jelzi a feldolgozás végét.
```

Természetesen a szófaj beolvasása a TSzófaj típus 'Be:' műveletével végzendő el!

- o a keresett szavakat tartalmazó text-file (\*.WRD):

```
Keresett szó:Szöveg
...
↵ Üres sor jelzi a feldolgozás végét.
```

#### A kimenet:

- o Rendezett szótár:

```
tárgyszó : célszó1 (szófaj1)
          : célszó2 (szófaj2)
          : ...
...
```

Ügyeljen a képernyő fenti „strukturáltságára”! Természetesen a szófaj kiírása a TSzófaj típus 'Ki:' műveletével végzendő el!



## ○ Keresett szavak fordítása:

Keresett szó : fordítás-szó <sub>1</sub> (szófaj <sub>1</sub> )
: fordítás-szó <sub>2</sub> (szófaj <sub>2</sub> )
: ...
...

Ügyeljen a képernyő fenti „strukturáltságára”! Természetesen a szófaj kiírása a TSzófaj típus ’Ki:’ műveletével végzendő el!

## 14. Borpince – *láncolt listák* szétválogatása

Egy borpincészet a borairól feljegyzi a következő adatokat: fajta (vörös, fehér, rozé), megnevezése, borvidék (Alföldi, Badacsonyi, Egri, Mátrai, Móri, Soproni, Tokaji, Villányi), évjárat, cukortartalom, szesztartalom.

Olvassa be egy *láncolt listába*, majd válogassa szét fajtánként külön-külön listába, *dinamikus*an. E listákat évjárat szerint növekvően *rendezi*, majd kiírja a képernyőre, lapozottan!

A *dinamikus*an határozóval arra utaltam, hogy amint egy borral végzett, máris megszünteti a bemeneti listában, s így mindig a lehető legkevesebb helyet igényli a programja.

Meghatározza, hogy az egyes fajtáknak melyik a legidősebb bora, továbbá, hogy mekkora az átlagos cukortartalma (szintén fajtánként)!

### Típus

```
TPincészet=Lista(TBor)
TBor=Rekord(fajta:TFajta,név:Szöveg,vidék:TVIDék,
             évjárat:Egész,cukor,sesz:Valós [tömegszázalék])
TFajta=(vörös,fehér,rozé)
TVIDék=(Alföldi,Badacsonyi,Egri,Mátrai,Móri,Soproni,
        Tokaji,Villányi)
```

A fenti felsorolás típusokat önálló modulokban meg kell valósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

### A bemenet:

- A borokat tartalmazó text-file (\*.BOR):

```
1. sor: Borok száma
2. sor: Fajta1 + Megnevezés1 + Vidék1 + Évjárat1 + vessző +
        Cukor1 + vessző + Szesz1
3. sor: Fajta2 + Megnevezés2 + Vidék2 + Évjárat2 + vessző +
        Cukor2 + vessző + Szesz2
...
```

### A kimenet:

- A borok fajták szerinti listája:

```
Fajta1
* Megnevezés11 , Vidék11
  Évjárat11 , Cukorfok11 , Szeszfok11
* Megnevezés12 , Vidék12
  Évjárat12 , Cukorfok12 , Szeszfok12
...
Fajta2
* Megnevezés21 , Vidék21
  Évjárat21 , Cukorfok21 , Szeszfok21
* Megnevezés22 , Vidék22
  Évjárat22 , Cukorfok22 , Szeszfok22
...
```

Ügyeljen az elválasztójelekre (csillag, vessző)!

- A borok fajták szerinti jellemzése (legidősb bor, átlag-cukorfok):

```
Fajta1 * legidősb bor megnevezése1 * átlagkor1
Fajta2 * legidősb bor megnevezése2 * átlagkor2
...
```

Ügyeljen az elválasztójelekre (csillag)!

## 15. Talált Tárgyak Osztálya – *láncolt listák* szétválogatása kategóriák szerint

A Talált Tárgyak Osztályán (TTO) az egyes tárgyakról feljegyezték a következőket: azonosítószám, kategória (ruházat, háztartási, szórakoztató, ékszer, irat, egyéb), megtalálás és visszaadás dátuma, érték. Ha nem jelentkezett még a gazdája érte, akkor a visszaadás dátuma: 0.0.0. Olvassa be *láncolt listák*ba a TTO nyilvántartását, majd *rendezze* kategóriák szerint! Ezután válogassa szét külön listákba az egyes kategóriákat! A rendezett kategória-listákat írja ki a képernyőre lapozottan! A szétválogatás több listára vonatkozik, ezért a részfeladatot *dinamikus*an kell elvégeznie.

A *dinamikus*an határozóval arra utaltam, hogy amint egy talált tárggyal végzett, máris szüntesse meg az eredeti listában, s így programja mindig a lehető legkevesebb helyet igényli.

Határozza, hogy hány tétel tartozik az egyes kategóriákba, s melyik a legértékesebb az egyes kategóriákban!

### Típus

TTTO=**Lista** (TTT)

TTT=**Rekord** (azonosító:Egész, kategória:TKategória, megtalálás, visszaadás:TDátum, érték:Egész)

TKategória= (Ruházat, Háztartási, Szórakoztató, Ékszer, Irat, Egyéb)

TDátum=**Rekord** (év, hó, nap:Egész)

A fenti felsorolás típusokat önálló modulokban meg kell valósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

### A bemenet:

- o A TTO nyilvántartását tartalmazó text-file (\*.TTO):

```
1. sor:  TTO tételek száma
2. sor:  azonosító1 + szóköz + érték1 + szóköz + kategória1
3. sor:  Mév1 pont Mhó1 pont Mnap1
4. sor:  Vév1 pont Vhó1 pont Vnap1
5. sor:  azonosító2 + szóköz + érték2 + szóköz + kategória2
6. sor:  Mév2 pont Mhó2 pont Mnap2
7. sor:  Vév2 pont Vhó2 pont Vnap2
...
```

### A kimenet:

- o A rendezett kategória-lista (jellemzőkkel):

```
Kategória1 # darabszám1
* Azonosító11 , Érték11
  Mév11 . Mhó11 . Mnap11 , Vév11 . Vhó11 . Vnap11
* Azonosító12 , Érték12
  Mév12 . Mhó12 . Mnap12 , Vév12 . Vhó12 . Vnap12
...
# Azonosító1leg1 , Érték1leg1
  Mév1leg1 . Mhó1leg1 . Mnap1leg1 , Vév1leg1 . Vhó1leg1 . Vnap1leg1
Kategória2 # darabszám2
* Azonosító21 , Érték21
  Mév21 . Mhó21 . Mnap21 , Vév21 . Vhó21 . Vnap21
* Azonosító22 , Érték22
  Mév22 . Mhó22 . Mnap22 , Vév22 . Vhó22 . Vnap22
...
```

```
# Azonosító2leg2 , Érték2leg2  
  Mév2leg2 . Mhó2leg2 . Mnap2leg2 . Vév2leg2 . Vhó2leg2 . Vnap2leg2  
  ...
```

A  $leg_1$  és a  $leg_2 \dots$  indexek az 1., a 2. ... kategória legértékesebb elemének indexét jelöli. Ügyeljen az elválasztójelekre (kettős kereszt, csillag, pont, vessző)! Üres kategóriát nem kell megjelenítenie.

**16.** Cégvezetők elfoglaltságainak kezelése *kétirányú, láncolt listával*, fontosság figyelembe vételével.

Egy cég néhány vezetőjének közeljövőbeli (két adott dátum közötti) elfoglaltságait az alábbi szerkezetű elemeket tartalmazó *kétirányú, láncolt listába* foglalva tároljuk:

```
Típus
TElfooglaltságotok=Lista (TElfooglaltságot)
TElfooglaltságotok=Rekord (
    érkiDő:TDátum [az igény érkezési ideje],
    Nap:TDátum, Mettől, Meddig:TIdő, Ki:TNév,
    Fontosság:TFontosság) ,
TDátum=Rekord (év, hó, nap:Egész)
TIdő=Rekord (óra, perc:Egész)
TFontosság= (VIP, Normál, Halasztható) .
```

Vezetőnként *külön-külön listát* kell vezetni a foglaltságról. A listában *érkiDő:TDátum*, *Nap:TDátum*, s azon belül *Mettől:TIdő* szerint rendezve vannak a foglaltsági adatok. **Módo-sítsa** a listákat új, file-ból jövő bejegyzésekkel. (Kezdetben üresek a listák.) Azaz ha eddig még szabad időpontra vonatkozó igény jött, illessze be; ha –részben vagy egészben– foglalt a megjelölt időintervallum, akkor „fontossága” szerint szüntesse meg az érintett összes foglaltságot, és helyettesítse az újjal, vagy utasítsa vissza az igényt. A fontossági sorrend a legfontosabbal kezdve: VIP..Halasztható. Az esetleg *visszautasítottakat* (vagy „*kitúrtakat*”) egy másik *listába* fűzze! Kérésre **jelentse meg** a heti elfoglaltságát egy adott személynek, és az addig hozzá érkezett, de visszautasított elfoglaltság-igénylések adatait is (igénylési adatok és azon partner neve, aki „megakadályozta” a foglalást, ha több ilyen is van, akkor is elég egyet).

```
Típus
TVisszautasítottok=Lista (TVisszautasított)
TVisszautasított=Rekord (
    Név:TNév, érkiDő:TDátum,
    Nap:TDátum, Mettől, Meddig:TIdő,
    Miatt:TNév [aki miatt lett visszautasítva])
```

**A bemenet:**

- o elfoglaltsági igénylések sorozata szöveges file-ban igénylési idő szerint rendezetten érkeznek – az igénylés-file egy eleme a file 7 egymásutáni sorában:

```
érkiDő:TDátum [amikor az igénylést beadták]
vezető:TVezetőNév
nap:TDátum [amikorra vonatkozik: tetszőleges!!!]
mettől:TIdő
meddig:TIdő
partner:TNév
fontosság:TFontosság
```

Természetesen a fent szereplő adatokat a megfelelő típus 'Be:'/'Ki:' műveleteivel kell beolvasni/kiírni, de feltételezhető a hibátlanság. Az eddig nem definiált típusról a következőket kell tudni:

```
TVezetőNév= ('Jó Isten', 'Luci Feri', 'Angyal Gábrriel')
[ez is felsorolástípus,
lehet persze más neveket is ide tenni]
```

**Példa – két igénylés az input-file-ból:**

```
2001 3 7
Luci Feri
2001 3 8
23 0
23 59
Belze Bub
VIP
```

Egy 2001. március 7-diki igénylés...

```
2001 3 17
Jó Isten
2001 3 9
12 15
16 30
Assisi Szent Ferenc
NORMÁL
```

... és egy másik ...

(-: *Nincs tévedés: egy korábbi dátumra vonatkozó kérésről van szó. Lassú volt a postás. :-)*

- Klaviatúra input: egy dátum és egy vezető neve: amikor, és aki elfoglaltsági állapotát, és az addig visszautasítottak listáját kéri kiírni.

**Egy példa – klaviatúra-input:**

```
2001 3 7
Luci Feri
```

Egy 2001. március 7. éjféle állapotkérés...

## 17. Szótár-vizsgálat – *láncolt listák* rendezése

A kétnyelvű szótárak –leegyszerűsítve– tekinthetők szópárok listájának, ahol az első szó a tárgynyelv (pl. angol) szava, a második pedig a célnyelvé (pl. magyar). Természetesen előfordulhat, hogy egy szónak több párja is van. Ez azt jelenti, hogy ugyanaz a szó több párosításban is szerepel. Feltehető, hogy a szó (mind a tárgy, mind a célnyelvi „oldalon”) egyetlen szóból áll, hossza nem haladja meg a 20 karaktert. Gyakran előfordulnak azonos alakú, de különböző szófajú szavak (pl. ír – ige, ír – jelző), ezek megkülönböztethetősége érdekében a szópár mellett közöljük a szófajt is.

Beolvasva a szópárokat, amelyeket az input két egymást követő sora tartalmaz, felépít egy *láncolt listát* (szótárat), majd rendezi a tárgynyelvi szó szerint növekvően. Írja ki a rendezett szótárat a képernyőre lapozottan, *táblázatosan!* Válaszoljon az alábbi kérdésekre:

- hány szócikket tartalmaz a szótár, ha egynek számítandók azok a párok, amelyben a tárgyszó megegyezik? (Például, ha előfordul a szótárban az (ír, writes) és az (ír, Irish) szópár, akkor ők együtt alkotnak egy szócikket: ír → writes, Irish.)
- melyik tárgyszónak van a legtöbb párja?

### Típus

```
TSzótár=Lista (TSzópár)
TSzópár=Rekord (tSzó, cSzó: Szöveg, szófaj: TSzófaj)
TSzófaj=(Ige, Főnév, Határozó, Jelző) [felsorolástípus]
```

### A bemenet:

- a szópárokat tartalmazó text-file (\*.DIC):

```
tárgyszó + szóköz + célszó: Szöveg
szófaj: TSzófaj
...
↓ Üres sor jelzi a feldolgozás végét.
```

Természetesen a szófaj beolvasása a TSzófaj típus 'Be:' műveletével végzendő el!

### A kimenet:

- Rendezett szótár (lapozottan, táblázatosan):

```
tárgyszó -> célszó (szófaj)
... ahány szópár van a szótárban
```

Ügyeljen arra, hogy a képernyőn a „->” jelek egymás alá essenek!

Természetesen a szófaj kiírása a TSzófaj típus 'Ki:' műveletével végzendő el!

- Egyéb szótárjellemzők:

```
szócikkek száma
a legtöbb párral rendelkező tárgyszó
```

**18.** Régészeti relikviák vizsgálata – *láncolt listák* értéklistája

Egy ásatáson gyűjtött tárgyakról feljegyzik: a fajtáját (korsó, tányér stb.), épségét (százalék), anyagát (ami bizonyos alapanyagok halmazát jelenti) és becsült korát. Egy tárgy értékét a következő formulával definiálják:  $kor * \text{épség}$ .

Hozzon létre egy érték szerint rendezett *láncolt listát*, amelyet kiír a képernyőre *rendezetten* és lapozottan!

**Típus**

```
TrTárgyak=Lista(TrTárgy)
TrTárgy=Rekord(fajta:TFajta,épség:Egész [százalék],
                anyag:TAnyag, kor:Egész)
TFajta=(Korsó,Tányér,Kés,Kard,Tór,Pohár,Egyéb)
TAnyag=Halmaz(TAlapanyag)
TAlapanyag=(Bronz,Vas,Réz,Agyag,Őn,Szövet,Más)
```

A fenti felsorolás és halmaz típusokat önálló modulokban kell valósítani. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

**A bemenet:**

- A tárgyakat tartalmazó text-file (\*.REL):

```
Tárgyak száma
Fajta1
Épség1
Anyag11 + vessző + Anyag12 + ...
Kor1
Fajta2
Épség2
Anyag21 + vessző + Anyag22 + ...
Kor2
...
```

**A kimenet:**

- A tárgyak érték szerinti listája:

```
Fajta1 = Épség1 , Kor1
          Anyag11 , Anyag12 +
Fajta2 = Épség2 , Kor2
          Anyag21 , Anyag22 +
...
```



## 19. Borpince – *láncolt listák* szétválogatása

Egy borpincészet a borairól feljegyzti a következő adatokat: fajta (vörös, fehér, rozé), megnevezése, borvidék (Alföldi, Badacsonyi, Egri, Mátrai, Móri, Soproni, Tokaji, Villányi), évjárat, cukortartalom, szesztartalom.

Olvassa be egy *láncolt listába*, majd válogassa szét borvidékenként külön-külön listába, *dinamikusán*. E listákat szesztartalom szerint *rendezi*, és kiírja a képernyőre, lapozottan!

A *dinamikusán* határozóval arra utaltam, hogy amint egy borral végzett, máris megtörli a bemeneti listából, s így folyamatosan a lehető legkevesebb helyigénnyel dolgozik a programja.

Meghatározza, hogy az egyes borvidékeknek melyik a legidősebb bora, továbbá, hogy milyen szélső határok között mozognak a borai szesztartalma (természetesen vidékenként)!

### Típus

```
TPincészet=Lista (TBor)
TBor=Rekord (fajta:TFajta,név:Szöveg,vidék:TVIDék,
             évjárat:Egész,cukor,esz:Valós [tömegszázalék])
TFajta=(vörös,fehér,rozé)
TVIDék=(Alföldi,Badacsonyi,Egri,Mátrai,Móri,Soproni,
        Tokaji,Villányi)
```

A fenti felsorolás típusokat önálló modulokban meg kell valósítania. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

### A bemenet:

- o A borokat tartalmazó text-file (\*.BOR):

```
1. sor: Borok száma
2. sor: Fajta1 + Megnevezés1 + Vidék1 + Évjárat1 + vessző +
        Cukor1 + vessző + Szesz1
3. sor: Fajta2 + Megnevezés2 + Vidék2 + Évjárat2 + vessző +
        Cukor2 + vessző + Szesz2
...
```

### A kimenet:

- o A borok fajták szerinti listája:

```
Vidék1
* Megnevezés11 , Fajta11
  Évjárat11 , Cukorfok11 , Szeszfok11
* Megnevezés12 , Fajta12
  Évjárat12 , Cukorfok12 , Szeszfok12
...
Vidék2
* Megnevezés21 , Fajta21
  Évjárat21 , Cukorfok21 , Szeszfok21
* Megnevezés22 , Fajta22
  Évjárat22 , Cukorfok22 , Szeszfok22
...
```

Ügyeljen az elválasztójelekre (csillag, vessző)!

- o A borok fajták szerinti jellemzése (legidősb bor, átlag-cukorfok):

```
Vidék1 * legidősb bor megnevezése1 * Min1 - Max1
Vidék2 * legidősb bor megnevezése2 * Min2 - Max2
...
```

Ügyeljen az elválasztójelekre (csillag, kötőjel)!

## 20. Talált Tárgyak Osztálya – *láncolt listák* szétválogatása

A Talált Tárgyak Osztályán (TTO) az egyes tárgyakról feljegyezték a következőket: azonosítószám, kategória (ruházat, háztartási, szórakoztató, ékszer, irat, egyéb), megtalálás és visszaadás dátuma, érték, és a megtaláló valamint a tulajdonos neve. Ha nem jelentkezett még a gazdája érte, akkor a visszaadás dátuma: 0.0.0, a tulajdonos: „ISMERTLEN”.

Olvassa be *láncolt listák*ba a TTO nyilvántartását, majd növekvően *rendezze* a tárgy TTO-n töltött idő szerint úgy, hogy ha még nem került vissza a gazdájához, akkor a mai napot kell visszaadás dátumnak tekinteni! Ezután válogassa szét két listába a visszaadottakat és a még vissza nem adottakat! A két listát írja ki a képernyőre lapozottan! A szétválogatást *dinamikus*an kell végeznie.

A *dinamikus*an határozóval arra utaltam, hogy amint egy talált tárggyal végzett, máris szüntesse meg az eredeti listában, s így programja mindig a lehető legkevesebb helyet igényli.

Határozza, hogy hány tételből állnak az egyes listák, továbbá: melyik a legértékesebb, még a TTO-n található talált tárgy!

### Típus

TTTO=**Lista** (TTT)

TTT=**Rekord** (azonosító: Egész, kategória: TKategória, megtalálás, visszaadás: TDátum, érték: Egész, megtaláló, tulajdonos: Szöveg)

TKategória= (Ruházat, Háztartási, Szórakoztató, Ékszer, Irat, Egyéb)

TDátum=**Rekord** (év: 1900..2005, hó: 1..12, nap: 1..31)

A fenti felsorolás típusokat önálló modulokban meg kell valósítani. Természetesen a fent szereplő adatokat a megfelelő típus 'Be:'/'Ki:' műveleteivel kell beolvasni/kiírni; feltételezhető a hibátlanság.

### A bemenet:

- o A TTO nyilvántartását tartalmazó text-file (\*.TTO):

```
1. sor: TTO tételek száma
2. sor: azonosító1 + szóköz + érték1 + szóköz + kategória1
3. sor: Mév1 pont Mhó1 pont Mnap1
4. sor: megtaláló1
5. sor: Vév1 pont Vhó1 pont Vnap1 + szóköz
6. sor: tulajdonos1
7. sor: azonosító2 + szóköz + érték2 + szóköz + kategória2
8. sor: Mév2 pont Mhó2 pont Mnap2
9. sor: megtaláló2
10. sor: Vév2 pont Vhó2 pont Vnap2
11. sor: tulajdonos2
...
```

### A kimenet:

- o A rendezett visszaadott és vissza nem adott tárgyak listája (jellemzőkkel):

```
Visszaadottak # darabszámva
* Azonosítóva 1 , Értékva 1
  Mévva 1 . Mhóva 1 . Mnapva 1 - Vévva 1 . Vhóva 1 . Vnapva 1
* Azonosítóva 2 , Értékva 2
  Mévva 2 . Mhóva 2 . Mnapva 2 - Vévva 2 . Vhóva 2 . Vnapva 2
...
```

```

# Azonosítóva leg , Értékva leg
Nem visszaadottak # darabszámnva
* Azonosítónva 1 , Értéknva 1
  Mévnva 1 . Mhónva 1 . Mnapnva 1 - Vévnva 1 . Vhónva 1 . Vnapnva 1
* Azonosítónva 2 , Értéknva 2
  Mévnva 2 . Mhónva 2 . Mnapnva 2 - Vévnva 2 . Vhónva 2 . Vnapnva 2
...

```

A <sub>leg ...</sub> indexek a még nem visszaadottak legértékesebb elmének indexét jelöli. Ügyeljen az elválasztójelekre (kettős kereszt, csillag, vessző, pont)! Üres kategóriát nem kell megjelenítenie.