

Számonkérés

Egy asztalban **tetszőleges** számú fiók lehet. Egy fiókot a megnevezése (maximum 10 db, angol ábécébe tartozó kisbetű) és a benne lévő tárgyak jellemznek. Egy fiókban maximum 3 tárgy van. Egy tárgynak csak a nevét tartjuk nyilván (max 20 db, angol ábécébe tartozó kisbetű). Feltehetjük, hogy egy asztalban legalább 3 fiók van, s mindegyik létező fiókban van minimum egy-egy tárgy, s az összes fiók és tárgy különbözik egymástól.

1. Alkossa meg az **asztal**, **fiok** és a **targy** típusokat, melyek alkalmasak a fentiek alapján adatok tárolására.
2. Adja meg a **fiok legKevesebb(asztal a)** függvényt, mely visszaadja a legkevesebb tárgyat tartalmazó fiókot! Ha több ilyen is van, elegendő az egyiket visszaadni!
3. Adja meg az **int ujTargy(asztal a, targy t)** függvényt, mely képes az **a** asztalba beletenni a **t** tárgyat. Abba a fiókba tegye, amelyikben a legkevesebb tárgy van, ha több ilyen is létezne, akkor mindegy, melyikbe teszi ezek közül. Amennyiben sikerült, egyet adjon vissza a függvény, különben nullát! A sikertelenség oka csak az lehet, ha már minden fiók tele van.
4. Adja meg az **int legtavolabbi(binfa b)** függvényt, mely megadja egy **int**-eket tartalmazó bináris fa gyökérétől a legtávolabbi elemet! Feltehetjük, hogy legalább egy elem van a fában és hogy a számok a $[0;100]$ intervallumba esnek, s csak egy ilyen elem van.
5. Implementálja C nyelven a következő algoritmust. Az egész számokat tartalmazó N elemű X és az M elemű Y és a Z tömb indexelése egytől kezdődik a pszeudokódban, ha nem muszáj, ne használjon indirekciós operátort a paraméterlistában:

```
Algoritmus zh(X, N, Y, M, Z, P)
    i:=N
    j:=M
    k:=1
    Amig (i>0 valamint j>0)
        Ha X[i] osztja Y[j]-t
            Z[k]:=X[i]
            k:=k+1
        Ha vege
            i:=i-1
            j:=j-2
    Amig vege
        P:=k
Algoritmus vege
```