**GB\_IN001\_2 Név: . . . . . . . . . . . . . . . . NEPTUN: . . . . . . . . . . . .Gép: . . . . . Jegy: . . Oktató aláírása: . . . . . . . . . . . . . . . . .**

Írjon C programot az alábbi feladatokra! A program indulásakor jelezze ki, hogy milyen feladatot végez el! Az input adatok bekérését és az eredmények közlését lássa el tájékoztató szövegekkel! A nem megfelelő input adat helyett mindenképpen és azonnal kérjen újat a program, esetleg hibajelzéssel!

 Írjon programot, amely egy osztály diákjait kezeli **láncolt lista** adatszerkezetben. A listaelemek mintájaként a következő típusdefiníció szolgáljon:

typedef struct szemelyLista {

int szemelyId;

char nev[50];

int \*baratok;

struct szemelyLista \*kov;

} szemely;

A láncolt listába egy szemely \*listaBeszur(szemely \*elso, char \*nev, int \*baratok) prototípusú függvény segítségével szúrjon adatokat, a szemelyId legyen egy minden sikeres adatbevitelnél eggyel növekvő érték, a \*baratok tömb végét -1 jelezze. A program indulásakor töltse fel az alábbi táblázat szerint értékekkel a láncolt listát. Ezután rendre kérje be üres sorig a neveket és a barátokat. A barátok létezését mindig ellenőrizze! Inputellenőrzésképp a név legyen minimum 4, maximum 49 betűs, a barát pedig valamilyen már létező **szám** (szemelyId). Az adatbevitel végén egy alkalommal írja ki listában tárolt neveket, és azok barátainak neveit is!

|  |  |  |  |
| --- | --- | --- | --- |
| **szemelyId** | **nev** | **baratok** | **barátok kiírva** |
| 0 | Ferenc | 1,2,4,-1 | Anna, Kata, Albert |
| 1 | Anna | 2,-1 | Kata |
| 2 | Kata | 1,-1 | Anna |
| 3 | Csilla | 0,1,2,4,-1 | Ferenc, Anna, Kata, Albert |
| 4 | Albert | 0,3,-1 | Ferenc, Csilla |

∑ **Értékelés**: Az alapfeladat megoldása 2 pont. Plusz illetve mínusz pont adandó a következőkért (*a teljesített pluszpont bekarikázandó*):

-1: Ha inputellenőrzés nincs vagy hibás.

-1: Ha a program fordítása során egyetlen, szabvány fejfájlok be nem kapcsolásából (#include) adódó, figyelmeztető üzenet is akad.

-1: Ha a programból történő kilépés előtt a lefoglalt memóriaterületeket nem szabadítja fel.

-1, -2: Ha az alapprogram működése bármiben is eltér a feladatban megfogalmazottól.

+1: Listázáskor csak a kölcsönös barátságokat jelzi. Kölcsönösnek tekintjük a barátságot, ha például a 0-ás felhasználó a 4-est tartja barátjának és a 4-es is a 0-ást.

+1: Kiírja, hogy ki a legnépszerűbb az osztályban, tehát kit tartanak a legtöbben barátjuknak és **nem**, hogy kinek van a legtöbb barátja.

+1: Amennyiben a program futása során név megadása helyett "T" karakter érkezik egy számmal (pl. "T4"), törli az adott személyt és a többi személynél a rá hivatkozó bejegyzéseket is.