**Minta csoportos kódolós ZH**

Írj programot, ami megoldja a következő feladatot. A programnak a következőket kell teljesítenie:

* követnie kell a megadott specifikációt és algoritmust: a megadott változóneveket használjátok (ékezeteket leszámítva), ha a feladat függvényt kér, akkor függvényt kell írni stb.
* ellenőrizni kell a bemenetet és kommunikálni kell a felhasználóval (pl. „Kérem az elemek számát”, „Nincs olyan ember, amilyet a feladat megadott” stb.). A hibás bemenetet (tehát ami nem felel meg az előfeltételnek) újra kell kérni. (Nem muszáj ékezeteket írni, illetve nem kell feltétlenül úgy formázni a kimenetet, ahogy azt a példában leírtam)
* a megoldás egyes részeit (deklarálás, beolvasás, a programozási tételek használata, kiírás) jól lehessen elkülöníteni. Minden adatot be kell olvasni és el kell tárolni.

**Fontos:** az olyan kód, ami nem fordul, automatikusan 0 pontot ér! (Úgyhogy ami hibás, azt inkább kommenteljétek ki, részpontot tudok adni rá).

**A feladat:**

Bájtországban n ember él (1 <= n <= 10000), mindegyiknek ismerjük a nevét (szöveg), életkorát (egész, 0 és 100 között van), barátainak a számát (egész, 0 és n-1 között van) és a foglalkozását (szöveg). Adjuk meg a választ a következő 2 kérdésre:

1. Hány különböző foglalkozás van Bájtországban?
2. Van-e olyan ember, akinek több barátja van, mint ahány éves?

A bemenet első sorában n szerepel, a következő n sorban pedig az egyes adatok: név, életkor, barátok száma és foglalkozás, szóközzel elválasztva. Pl:

**Bemenet:** **Kimenet:**

4 3 különböző foglalkozás van Bájtországban

Bájtika 40 3 orvos Nincs ilyen ember

Pötyike 15 0 kertész

Marika 68 2 orvos

Móricka 25 0 mérnök

**Specifikáció:**

Be: n ∈ N, lakók ∈ Lakó[1..n]

Lakó = Név x Kor x Barátok x Foglalkozás

Név = S, Kor = Z, Barátok = Z, Foglalkozás = S

Ki: foglalkozásDb ∈ N, van ∈ L

Ef: 1 <= n <= 10000 és (∀ i ∈ [1..n]: 0 <= lakók[i].kor <= 100

és 0 <= lakók[i].barát <= n - 1)

Uf: foglalkozásDb = DARAB(i = 1..n, nemVoltMég(i)) és

van = VAN(i = 1..n, lakók[i].kor < lakók[i].barátok)

Fv: nemVoltMég: N -> L

nemVoltMég(ind) = nem VAN(j = 1..ind - 1,lakók[j].foglalkozás = lakók[ind].foglalkozás)

**Visszavezetés:**

1. részfeladat:

megszámlálás:

e..u ~ 1..n

T(i) ~ nemVoltMég(i)

db ~ foglalkozásDb

eldöntés:

e..u ~ 1..ind - 1

T(j) ~ lakók[j].foglalkozás = lakók[ind].foglalkozás

1. részfeladat:

eldöntés

e..u ~ 1..n

T(i) ~ lakók[i].kor < lakók[i].barátok

**Algoritmus:**

1. részfeladat:

Fő algoritmus:

A white box with black text and numbers

Description automatically generated

Függvény:

A screenshot of a computer

Description automatically generated

1. részfeladat:

Fő algoritmus:

A white background with black text

Description automatically generated