**A**

Egy szöveges állományban egy cég gépjárműveinek tankolásainál felvett üzemanyag mennyiségét tároljuk literben. Az állomány minden sorában egy-egy gépjármű adatait találjuk: a jármű rendszámát és mellette annyi darab 0 és 80 közé eső számot, ahányszor tankoltak a járműbe. A tankolások száma járművenként eltérő, sőt olyan jármű is lehet, amelybe egyáltalán nem tankoltak. Az adatok között szóköz(ök) és/vagy tabulátor jelek találhatók. Feltehetjük, hogy az állomány helyesen van kitöltve.

Mondja meg, hogy melyik járműbe tankoltak legkevesebbszer, valamint listázza ki azon járművek rendszámát, amelybe összesen 300 liternél többet tankoltak!

**Például az input fájl adatai:**

CEG-001 15 23 45 7

CEG-003 10 34 42

CEG-101 9 10 10 8 19 47

Megfelelt szintű (közepes) a megoldás akkor, ha

* az érvényes tesztesetekre jól működik a program,
* a program ciklusai tanult programozási tételekből származnak,
* a program a szöveges állománynak legfeljebb egy sornyi adatát tárolja egyszerre.

Kiválóan megfelelt szintű a megoldás akkor, ha megfelelt, és

* az érvénytelen teszteseteket is kezeli: nem létező állománynév észlelése,
* nem ágyaz közvetlenül ciklust ciklusba, hanem ezt elkerülendő megfelelően paraméterezett függvényeket használ,
* egy jármű adatainak az input fájlból történő beolvasására egy külön függvény szolgál. (Ezt elkészítheti úgy is, hogy egy járműről csak annak rendszámát, a tankolásainak számát és a tankolt üzemanyag összegét adja vissza.)

Értékelés után a programját EHAkód.zip formában az alábbi helyre töltse fel:

[\\inf.elte.hu\dfs\zh\PROG\E](file:///\\inf.elte.hu\dfs\zh\PROG\E)

**B**

Egy szöveges állományban egy cég gépjárműveinek tankolásainál felvett üzemanyag mennyiségét tároljuk literben. Az állomány minden sorában egy-egy gépjármű adatait találjuk: a jármű rendszámát és mellette annyi darab 0 és 80 közé eső számot, ahányszor tankoltak a járműbe. A tankolások száma járművenként eltérő, sőt olyan jármű is lehet, amelybe egyáltalán nem tankoltak. Az adatok között szóköz(ök) és/vagy tabulátor jelek találhatók. Feltehetjük, hogy az állomány helyesen van kitöltve.

Mondja meg, hány olyan jármű van, amelybe egyáltalán nem tankoltak, valamint listázza ki azon járművek rendszámát, amelybe átlagosan 20 liternél több került!

**Például az input fájl adatai:**

CEG-001 15 23 45 7

CEG-003 10 34 42

CEG-101 9 10 10 8 19 47

Megfelelt szintű (közepes) a megoldás akkor, ha

* az érvényes tesztesetekre jól működik a program,
* a program ciklusai tanult programozási tételekből származnak,
* a program a szöveges állománynak legfeljebb egy sornyi adatát tárolja egyszerre.

Kiválóan megfelelt szintű a megoldás akkor, ha megfelelt, és

* az érvénytelen teszteseteket is kezeli: nem létező állománynév észlelése,
* nem ágyaz közvetlenül ciklust ciklusba, hanem ezt elkerülendő megfelelően paraméterezett függvényeket használ,
* egy jármű adatainak az input fájlból történő beolvasására egy külön függvény szolgál. (Ezt elkészítheti úgy is, hogy egy járműről csak annak rendszámát, a tankolásainak számát és a tankolt üzemanyag összegét adja vissza.)

Értékelés után a programját EHAkód.zip formában az alábbi helyre töltse fel:

[\\inf.elte.hu\dfs\zh\PROG\E](file:///\\inf.elte.hu\dfs\zh\PROG\E)

**C**

Egy vállalat portaszolgálatán feljegyzik a belépő és kilépő személyek adatait. Belépéskor rögzítik az aktuális időpontot, a belépő nevét és azonosítására szolgáló igazolványszámot, távozáskor pedig a fenti három adat mellé kerül a kilépés időpontja. A napi adatokat belépési idő szerint növekvően rendezve egy szöveges állományban tárolják.

Például az input fájl adatai:

08:30 Gipsz\_Jakab 1234567A 10:15

08:35 Gyanús\_Móric 1x1x1x1x1x

08:45 Szer\_Elek 11112222 09:15

Feltehetjük, hogy az állomány helyesen van kitöltve, annak egy sorában három vagy négy sztringet találunk, amelyeket tetszőleges számú elválasztó jel határol, és ezeknek a sztringeknek a belsejében viszont nem találunk elválasztó jelet.

Gyűjtse ki azon személyek nevét és azonosítóját, akik jelenleg is benn tartózkodnak a vállalat területén (beléptek, de még nem távoztak), valamint a távozók közül adja egy olyan személy nevét, aki a leghosszabb ideig tartózkodott benn!

Megfelelt szintű (közepes) a megoldás akkor, ha

* az érvényes tesztesetekre jól működik a program,
* a program ciklusai tanult programozási tételekből származnak,
* a programban nem használhat olyan tömböt vagy sztringet, amelynek hossza a szöveges állomány sorainak számától függ,
* a program csak egyszer olvassa végig az állományt.

Kiválóan megfelelt szintű a megoldás akkor, ha megfelelt, és

* az érvénytelen teszteseteket is kezeli: nem létező állománynév észlelése, üres sorok az állományban nem okoznak problémát,
* nem ágyaz közvetlenül ciklust ciklusba, hanem ezt elkerülendő megfelelően paraméterezett függvényeket használ,
* a szekvenciális inputfájl egy sorának beolvasására olyan függvényt használ, amely csak a feldolgozáshoz szükséges adatokat (kilépett-e illetve mennyi ideig volt benn) adja vissza.

Értékelés után a programját EHAkód.zip formában az alábbi helyre töltse fel:

[\\nas1.inf.elte.hu\zh\PROG\E](file:///\\nas1.inf.elte.hu\zh\PROG\E)

**D**

Egy szavazást több körzetben bonyolítottak le, ahol a jelöltek száma *m* volt. Egy szöveges állományban rögzítették a szavazás eredményét. Egy sorban szerepel az egyes körzetek eredménye: a körzet száma (pozitív egész), a szavazásra jogosultak száma (természetes szám), sorban egymás után az *m* jelöltre leadott szavazatok száma (természetes szám) és az érvénytelen szavazatok száma (természetes szám). Az adatokat tetszőleges számú elválasztó jel (szóköz, tabulátor jel) választhatja el-

Írjon programot, amely megadja, hogy érvényes volt-e a szavazás (azaz a jogosultaknak több, mint fele szavazott-e), illetve mely körzetben voltak legaktívabbak a választók, azaz hol mentek el a legnagyobb százalékban szavazni!

Minta adatok *m*=5 esetén. Az *m* értékét az állomány legelső sora tartalmazza:

5

1 1548 342 359 611 103 28 3

2 2034 148 345 1290 68 158 8

3 925 345 63 384 30 102 0

4 1246 156 48 984 10 15 12

5 1950 220 212 831 412 240 2

6 3435 512 342 1215 548 330 5

7 4012 1500 136 1450 445 52 18

8 2320 366 257 1190 142 357 3

9 3100 366 155 1402 302 312 4

10 2699 148 338 1352 250 248 5

(*körzet jogosult 1. 2. 3. 4. 5. érvénytelen*)

Megfelelt szintű (közepes) a megoldás akkor, ha

* az érvényes tesztesetekre jól működik a program,
* a program ciklusai tanult programozási tételekből származnak,
* a programban nem használhat olyan tömböt vagy sztringet, amelynek hossza a szöveges állomány sorainak számától függ,
* a program csak egyszer olvassa végig az állományt.

Kiválóan megfelelt szintű a megoldás akkor, ha megfelelt, és

* az érvénytelen teszteseteket is kezeli: nem létező állománynév észlelése; üres sorok az állományban ne okoznak problémát; hibás adatszolgáltatású körzetek (többen mentek szavazni, mint ahányan jogosultak) figyelmen kívül hagyása,
* nem ágyaz közvetlenül ciklust ciklusba, hanem ezt elkerülendő megfelelően paraméterezett függvényeket használ,
* a szekvenciális inputfájl egy sorának beolvasására olyan függvényt használ, amely csak a feldolgozáshoz szükséges adatokat (kilépett-e illetve mennyi ideig volt benn) adja vissza.

Értékelés után a programját EHAkód.zip formában az alábbi helyre töltse fel:

[\\nas1.inf.elte.hu\zh\PROG\E](file:///\\nas1.inf.elte.hu\zh\PROG\E)