

Név: _____

Programozás 2. ZH (A szakirány) - A. csoport

Egy több napos magasugró verseny lebonyolításában szeretnék a segítségedet kérni. A konkrét probléma az lenne, hogy egy napon még meg tudják oldani, hogy sorbarendezi az eredményeket és felírják egy táblázatba, de több nap alatt elég sok információ gyűlik össze. A megrendelő megelégszik azzal is, hogy 2 fájlt tudjon összevonni, mivel ha 2-t össze tud, akkor akármennyit össze tud.

Bemenő adatok

A bemenő adatok a versenyzők által teljesített ugrások, ahol a név (szóköz nélküli alfabetaikus) teljes mértékben azonosítja az adott versenyzőt **(név szerint van rendezve az input)**. A magasság méterben van megadva. A nemzetiség szimplán csak másolódik az outputra.

Anthon 3 1.66 1.77 1.55 RU	[Név] [Ugrási magasságok] [Nemzetiség]
Bela 1 1.56 HU	[Név] [Ugrási magasságok] [Nemzetiség]
John 2 1.75 1.55 EN

Anthon 1 1.78 RU
Bela 2 1.76 1.56 HU
John 1 1.12 EN
Wiliam 3 1.82 1.65 1.44 EN

Előfeltételek:

- Ugrások száma pozitív egész
- Nemzetiség 2 karakter

[Ugrási magasságok] = [Ugrási magasságok száma] [1. ugrási magasság] ... [n. ugrási magasság]

Kimenő adatok

A cél az lenne, hogy összefuttassuk az eredményeket úgy, hogy ha mind a két fájlban megtalálható az adott versenyző, akkor az kerül be a kimenetbe, amelyikben a maximum ugrott magasság nagyobb.

- Ha azonos a maximum, akkor teljesen mindegy melyiket választjuk.
- A kimenetre a teljes rekordot ki kell írunk.

Anthon 1 1.78 RU	Anthon első bemenetén 1.77 a maximum a második bemeneten
Bela 2 1.76 1.56 HU	1.78, így a 2. fájlból kerül be Anthon rekordja...
John 2 1.75 1.55 EN	
Wiliam 3 1.82 1.65 1.44 EN	

Követelmény

- Fájlból/ba kell beolvasni/kiírni az adatokat, az órán tanult szekvenciális input/output osztályokkal.
- A versenyző tárolására létre kell hozni egy osztályt, amiben:
 - Kiíró és beolvasó operátort kell megvalósítani barát függvény segítségével.
 - A versenyző osztály dobjon exception-t ha valami előfeltétel nem megfelelő.
- Figyeljünk a kódolási konvenciókra és a modulokra/függvényekre bontás szabályaira.
- Az adatok privátok legyenek, ha szükséges az elérésük/módosításuk megfelelő setter/getter-t kell létrehozni hozzá.
- Szükséges osztályok: SzekIn, SzekOut, Rekord
- A program ciklusai, csak a tanult tételből származhatnak.
- Közepes szint: kivétel kezelés nem kell, rekord osztályt elég struct-tal megvalósítani, barát függvény nem kell.

Név: _____

Programozás 2. ZH (A szakirány) - B. csoport

Megbízónk felmérést készít a legolcsóbb telefonokról. Több ember végezte a felmérést és típusnév szerinti sorrendben feljegyezte a megadott modellek árait (ezer forintban) az általa vizsgált boltokban. A megrendelőnk úgy szeretné, hogy ha több felmérő is vizsgálat alá vette az egyik típust, akkor a kimenetbe az kerüljön be, aki több helyen talált 5000 Forint alatti árat.

Bemenő adatok

A bemenő adatok a felmérések, ahol a típusnév (szóköz nélküli alfabetaikus) teljes mértékben azonosítja az adott modellt (**típusnév szerint van rendezve az input**). Ezen kívül még a vizsgáló monogramja található meg egy rekordban, ami szintén szóköz nélküli alfabetaikus. **Az ár ezer forintban van megadva**. A monogram csak azért kell, hogy azonosítani tudjuk, hogy ki találta az adott árakat.

GALAXY 3 10 5 2 RP	[Típusnév] [Vizsgálatok] [Vizsgáló monogramja]
NOC5 1 10 RP	[Típusnév] [Vizsgálatok] [Vizsgáló monogramja]
SENEO 2 2 4 ME

GALAXY 2 2 3 AA
H1 2 1 10 AA
RBM 1 10 HG
SENEO 3 2 10 6 CV

Előfeltételek:

- Az ár természetes szám
- Monogram 2 karakter

[Vizsgálatok] = [Vizsgálatok száma] [1. vizsgált boltban az ár] ... [n. vizsgált boltban az ár]

Kimenő adatok

A cél az lenne, hogy összefuttassuk a felméréseket úgy, hogy ha mind a két fájlban megtalálható az adott telefon, akkor az kerül be a kimenetbe, amelyik többször talált 5000 Forintos ár alatti árat.

- Ha azonos az 5000 alatti ár találat, akkor mindegy melyiket választjuk.
- A kimenetre a teljes rekordot ki kell írunk.

GALAXY 2 2 3 AA
H1 2 1 10 AA
NOC5 1 10 RP
RBM 1 10 HG
SENEO 2 2 4 ME

A galaxy telefon esetében AA felmérési eredménye került be a kimeneti fájlba, mert ő 2 helyen(2 3) talált 5000 alatti árat, amíg RP csak 1 helyen(10 5 2)...

Követelmény

- Fájlból/ba kell beolvasni/kiírni az adatokat, az órán tanult szekvenciális input/output osztályokkal.
- A versenyző tárolására létre kell hozni egy osztályt, amiben:
 - Kíró és beolvasó operátort kell megvalósítani barát függvény segítségével.
 - A telefon osztály dobjon exception-t ha valami előfeltétel nem megfelelő.
- Figyeljünk a kódolási konvenciókra és a modulokra/függvényekre bontás szabályaira.
- Az adattagok privátok legyenek, ha szükséges az elérésük/módosításuk megfelelő setter/getter-t kell létrehozni hozzá.
- Szükséges osztályok: SzekIn, SzekOut, Rekord
- A program ciklusai, csak a tanult tételből származhatnak.
- Közepes szint: kivétel kezelés nem kell, rekord osztályt elég struct-tal megvalósítani, barát függvény nem kell.