**A**

Egy szekvenciális inputfájlban egy mozi egy napi vetítései és forgalma található. A fájl egy eleme egy eladott jegy adatait tartalmazza szóközökkel tagolva: a vetített film címét, a terem nevét (ez szóközt nem tartalmaz), a vetítés kezdetének idejét (HH:MM formában), és a vásárolt hely számából és az eladott jegy típusából álló érték-párt. Három különböző típusú jegyet lehet vásárolni: gyermek (G) – 500 forint, diák (D) – 800 forint, felnőtt (F) – 1000 forint. A fájl sorai kezdési időpontok szerint rendezettek. Egy adott időpontban legfeljebb egy film vetítése kezdődik. Például:

cím terem idő hely típus

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 9 G

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 3 G

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 5 F

A vámpírrá lett zombi vérfarkas Coppola 19:45 15 F

A vámpírrá lett zombi vérfarkas Coppola 19:45 7 D

***Megfelelt szint*** (közepes): Itt feltételezheti, hogy a szöveges állományban a vetített film címe helyén egy 6 karakterből álló azonosító áll (amelyik elválasztó jeleket nem tartalmaz). Melyik vetítés hozta a legnagyobb bevételt?

***Kiválóan megfelelt szint*** (jeles): Itt a szöveges állomány a vetített film azonosítója helyett a film címét tartalmazza (és ebben elválasztó jelek is lehetnek). A megfelelt szint válasza mellett azt is adja meg, hogy volt-e olyan vetítés, amelyikre csak diákok váltottak jegyet.

A programja mindkét szinten elégítse ki az alábbi követelményeket:

* az érvényes tesztesetekre helyesen működik (itt felteheti, hogy az állomány minden sora a megadott módon van kitöltve), valamint az üres fájl és a nem létező fájl esetét is lekezeli;
* a program ciklusai tanult programozási tételekből származnak, ezek nevét kommentben feltünteti (ha rekurzív függvényt használ, azt definiálja);
* a szöveges állományt csak egyszer olvashatja végig;
* nem használhat a szöveges állomány sorainak számától függő méretű ideiglenes tárolót (tömböt, sztringet);
* egy osztályt készít annak a felsorolónak a típusára, amellyel a feladat megoldását legfelső szinten tervezi.

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

Windows alól: [\\nas2.inf.elte.hu\zh\PROG](file:///\\nas2.inf.elte.hu\zh\PROG)

Linux alól: **smb://nas2.inf.elte.hu** a **zh\PROG** könyvtárba

**B**

Egy szekvenciális inputfájlban egy mozi egy napi vetítései és forgalma található. A fájl egy eleme egy eladott jegy adatait tartalmazza szóközökkel tagolva: a vetített film címét, a terem nevét (ez szóközt nem tartalmaz), a vetítés kezdetének idejét (HH:MM formában), és a vásárolt hely számából és az eladott jegy típusából álló szám-karakter értékpárt. Három különböző típusú jegyet lehet vásárolni: gyermek (G) – 500 forint, diák (D) – 800 forint, felnőtt (F) – 1000 forint. A fájl sorai kezdési időpontok szerint rendezettek. Egy adott időpontban legfeljebb egy film vetítése kezdődik. Például:

cím terem idő hely típus

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 9 G

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 3 G

Nyiszi nyuszi és a locsi-fecsi tönk Jancsó 10:15 5 F

A vámpírrá lett zombi vérfarkas Coppola 19:45 15 F

A vámpírrá lett zombi vérfarkas Coppola 19:45 7 D

***Megfelelt szint*** (közepes): Itt feltételezheti, hogy a szöveges állományban a vetített film címe helyén egy 6 karakterből álló azonosító áll (amelyik elválasztó jeleket nem tartalmaz). Melyik vetítésen vett részt a legtöbb gyerek?

***Kiválóan megfelelt szint*** (jeles): Itt a szöveges állomány a vetített film azonosítója helyett a film címét tartalmazza (és ebben elválasztó jelek is lehetnek). A megfelelt szint válasza mellett azt is adja meg, hogy vett-e minden vetítésre jegyet diák?

A programja mindkét szinten elégítse ki az alábbi követelményeket:

* az érvényes tesztesetekre helyesen működik (itt felteheti, hogy az állomány minden sora a megadott módon van kitöltve), valamint az üres fájl és a nem létező fájl esetét is lekezeli;
* a program ciklusai tanult programozási tételekből származnak, ezek nevét kommentben feltünteti (ha rekurzív függvényt használ, azt definiálja);
* a szöveges állományt csak egyszer olvashatja végig;
* nem használhat a szöveges állomány sorainak számától függő méretű ideiglenes tárolót (tömböt, sztringet);
* egy osztályt készít annak a felsorolónak a típusára, amellyel a feladat megoldását legfelső szinten tervezi.

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

Windows alól: [\\nas2.inf.elte.hu\zh\PROG](file:///\\nas2.inf.elte.hu\zh\PROG)

Linux alól: **smb://nas2.inf.elte.hu** a **zh\PROG** könyvtárba

**C**

Egy tömegközlekedéssel foglalkozó vállalat bevezette az elektronikus bérleteket tesztelési céllal egy napra, és egy szöveges fájl tartalmazza az eltárolt adatokat: egy sorban az utas nevét, az utas azonosítóját (8 jegyű szám), és egy időkód (HH:MM) – járat (sztring) párt. A fájl azonosítók szerint rendezett. Például:

Munkába Járó András 12345678 6:30 9

Munkába Járó András 12345678 7:45 17A

Munkába Járó András 12345678 7:36 57

***Megfelelt szint*** (közepes): Itt feltételezheti, hogy a szöveges állományban nem szerepel az utas neve, csak az azonosítója. Átlagosan hányszor utazott egy utas?

***Kiválóan megfelelt szint*** (jeles): Itt a szöveges állomány egy sora az utas nevével kezdődik, amely többtagú is lehet, és ezt követi az azonosítója. A megfelelt szint válasza mellett azt is adja meg, hogy vajon utazott-e minden utas a 7E járattal.

A programja mindkét szinten elégítse ki az alábbi követelményeket:

* az érvényes tesztesetekre helyesen működik (itt felteheti, hogy az állomány minden sora a megadott módon van kitöltve), valamint az üres fájl és a nem létező fájl esetét is lekezeli;
* a program ciklusai tanult programozási tételekből származnak, ezek nevét kommentben feltünteti (ha rekurzív függvényt használ, azt definiálja);
* a szöveges állományt csak egyszer olvashatja végig;
* nem használhat a szöveges állomány sorainak számától függő méretű ideiglenes tárolót (tömböt, sztringet);
* egy osztályt készít annak a felsorolónak a típusára, amellyel a feladat megoldását legfelső szinten tervezi.

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

Windows alól: [\\nas2.inf.elte.hu\zh\PROG](file:///\\nas2.inf.elte.hu\zh\PROG)

Linux alól: **smb://nas2.inf.elte.hu** a **zh\PROG** könyvtárba

**D**

Egy tömegközlekedéssel foglalkozó vállalat bevezette az elektronikus bérleteket tesztelési céllal egy napra, és egy szöveges fájl tartalmazza az eltárolt adatokat: egy sorban az utas nevét, az utas azonosítóját (8 jegyű szám), és egy időkód (HH:MM) – járat (sztring) párt. A fájl azonosítók szerint rendezett. Például:

Munkába Járó András 12345678 6:30 9

Munkába Járó András 12345678 7:45 17A

Munkába Járó András 12345678 7:36 57

***Megfelelt szint*** (közepes): Itt feltételezheti, hogy a szöveges állományban nem szerepel az utas neve, csak az azonosítója. Hány olyan utas volt, aki utazott a 7E járattal?

***Kiválóan megfelelt szint*** (jeles): Itt a szöveges állomány egy sora az utas nevével kezdődik, amely többtagú is lehet, és ezt követi az azonosítója. A megfelelt szint válasza mellett azt is adja meg, hogy volt-e olyan utas, aki legalább ötször utazott.

A programja mindkét szinten elégítse ki az alábbi követelményeket:

* az érvényes tesztesetekre helyesen működik (itt felteheti, hogy az állomány minden sora a megadott módon van kitöltve), valamint az üres fájl és a nem létező fájl esetét is lekezeli;
* a program ciklusai tanult programozási tételekből származnak, ezek nevét kommentben feltünteti (ha rekurzív függvényt használ, azt definiálja);
* a szöveges állományt csak egyszer olvashatja végig;
* nem használhat a szöveges állomány sorainak számától függő méretű ideiglenes tárolót (tömböt, sztringet);
* egy osztályt készít annak a felsorolónak a típusára, amellyel a feladat megoldását legfelső szinten tervezi.

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

Windows alól: [\\nas2.inf.elte.hu\zh\PROG](file:///\\nas2.inf.elte.hu\zh\PROG)

Linux alól: **smb://nas2.inf.elte.hu** a **zh\PROG** könyvtárba