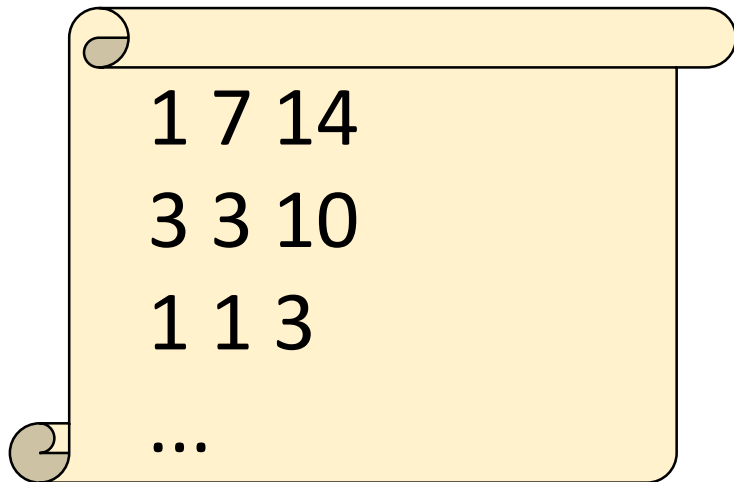


Egy futárszolgálat a futárjainak a megtett utak alapján ad fizetést. Az egyik futár egy héten át feljegyezte fuvarjai legfontosabb adatait, és azokat eltárolta egy állományban

A fájlban minden sor egy-egy út adatait tartalmazza egymástól szóközzel elválasztva.

- nap sorszáma, ami 1 és 7 közötti érték lehet
- a napon belüli fuvarszám,
- az adott fuvar során megtett utat jelenti kilométerben, egészre kerekítve



1 7 14  
3 3 10  
1 1 3  
...



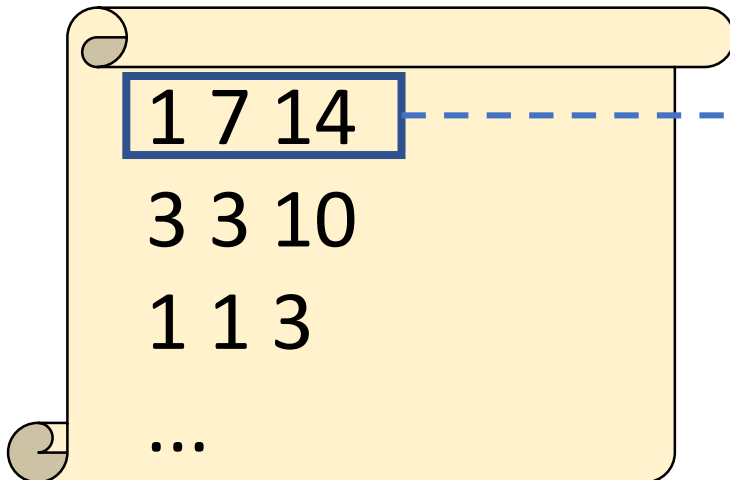
AppFutár :class főprogram

**f1():** Olvassa be az adatok.txt állományban talált adatokat.

tétel: másolás

A szokásos módon egy sor kiolvasása és a konstruktor hívása.

A szeletelés a *class* feladata!



<Ut> utak :lista

**Ut** :class

- int nap
- int fuvar
- int km

+ Ut(String sor)

- + getNap
- + getFuvar
- + getKm



**f2():** Kérjen be a felhasználótól egy tetszőleges napot és fuvarszámot.

Írja ki a képernyőre, ennek a fuvarnak az adatait!

**tétel:** lineáris keresés

be: (*nap*)

be: (*fuvarszám*)

*i*=0

ciklus amíg (*i*<*N* és !(getNap()*=nap* és a getFuvar()*=fuvarszám*))

*i*++

ciklus vége

ha *i*<*N* akkor ki: (*nap*, *fuvarszám*, getKm())

    különben ki: („nincs ilyen adat”)

ha vége

**Ut** :class

- int nap
- int fuvar
- int km

+ Ut(String sor)

+ getNap

+ getFuvar

+ getKm



**f3():** Tudjuk, hogy a futár minden héten tart legalább egy szabadnapot.

Írja ki a képernyőre, hogy a hét hányadik napjain nem dolgozott a futár!

**tétel:** kiválogatás tömbbe

```
új napok[8]
ciklus (i=0; i<N; i++)
|      k = getNap() //1..7
|      napok[k]++
ciklus vége
ciklus (i=1; i<8; i++)
|      ha napok[i] = 0
|      |      akkor ki: (i)
|      ha vége
ciklus vége
```

**Ut** :class

- int nap
- int fuvar
- int km

+ **Ut**(String sor)

- + getNap
- + getFuvar
- + getKm



**f4():** Írja ki a képernyőre, hogy a hét melyik napján volt a legtöbb fuvar! Amennyiben több nap is azonos, maximális számú fuvar volt, elegendő ezek egyikét kiírnia.

**tétel:** kiválogatás tömbbe, max kiválasztás

```
új napok[8]
ciklus (i=0; i<N; i++)
|       napok[getNap()]++ //1..7
|
ciklus vége
```

```
maximum kiválasztás(napok[maxIndex])
ki: (maxIndex)
```

**Ut** :class

- int nap
- int fuvar
- int km

+ **Ut**(String sor)

- + getNap
- + getFuvar
- + getKm



**f5():** Számítsa ki és írja a *km.txt* állományba, hogy az egyes napokon hány kilométert kellett tekerni!

**tétel:** összegzés *tömbbe*

```
új napok[8]
ciklus (i=0; i<N; i++)
|      napok[getNap()] = napok[getNap()] + getKm()
|
ciklus vége

ciklus (i=1; i<8; i++)
|      fájlba: (napok[i])
|
ciklus vége
```

**Ut** :class

- int nap
- int fuvar
- int km

+ **Ut**(String sor)

- + getNap
- + getFuvar
- + getKm



**f6():** A futár az egyes fuvarokra az út hosszától függően sávosan kap fizetést az alábbi szerint:

500 Ft	700 Ft	900 Ft	1400 Ft	2000 Ft
1 – 2 km	3 – 5 km	6 – 10 km	11 – 20 km	21 – 30 km

Kérjen be a felhasználótól egy tetszőleges napot és fuvarszámot, és határozza meg, hogy mennyi Ft jár érte!

**tétel:** lineáris keresés

Keresés ugyanúgy mint az f2()-ben...

```
ha i<N akkor ki: (getFt())
|      különben ki: („nincs ilyen adat”)
|
ha vége
```

**Ut** :class

- int nap
- int fuvar
- int km
- int Ft // tbl-ből

+ **Ut**(String sor)

- + getNap
- + getFuvar
- + getKm
- + getFt

