

Az `eladott.txt` állomány Budapest-Répcelak út jegyvásárlásait tartalmazza.

A vonal teljes hossza 172 km, és minden megkezdett 10 km után fizetendő összeg 71 Ft.

A jegy árának meghatározásakor az értéket öttel osztható számra kell kerekíteni.

40 0 62 [ülés, mettől, meddig]

*A megtett távolság 62 km, tehát 6+1 darab 10 km hosszú szakaszért kell fizetnie, ennek értéke  $7 \cdot 71$ , azaz 497 Ft.*

*Mivel kerekíteni kell, ezért a fizetendő összeg 495 Ft*

1. Olvassa be az *eladott.txt* szóközökkel tagolt szövegfájlt a **<Utas>** *utasok* listába.
2. A legutolsó jegyvásárló sorszáma, ülésének száma és az általa beutazott távolság.
3. Hol ültek akik végig utazták a teljes utat?
4. Sorszám, megtett út és mennyit fizetett az, aki többet utazott mint 100 km.
5. A jegyekből mennyi bevétele származott a társaságnak

Olvassa be a *szavazatok.txt* szóközzel tagolt szövegfájlt a ----->  
<Utas> *utasok* listába.

pl.: 7 115 156

<Utas> *utasok* :lista  
másolás-tétele

Utas osztály

- szék :egészszám
- mettől :egészszám
- meddig :egészszám

A főprogramban előltesztelő ciklussal fájl beolvasása és az **Utas** osztály konstruktorának hívása, aki feltölti a 3 adatmezőt ( *tmp[ ]...* )

A legutolsó jegyvásárló sorszáma,  
ülésének száma és az általa  
beutazott távolság



<Utas> *utasok* :lista  
N. elem kiírása

**Utas** osztály

- szék :egészszám
- mettől :egészszám
- meddig :egészszám
- km :egészszám

+ getSzék

+ getKm

A főprogramban utasok létszáma ismert,  
az **Utas** osztály **getSzék** és **getKm** metódusokkal kell lekérdezni

Hol ültek akik végig utazták a teljes  
utat? ----->

<Utas> *utasok* :lista

**kiválogatás tétele ( getVégig)**

**Utas** osztály

- szék :egészszám
- mettől :egészszám
- meddig :egészszám
- km :egészszám
- végig :logikai

+ getSzék

+ isVégig

A főprogramban végig az utasokon  
ha isVégig = igaz  
akkor: ki (**getSzék**)

Sorszám, megtett út és mennyit  
fizetett az, aki többet utazott mint **100 km**.  
----->

A főprogramban végig az utasokon  
ha `getKm > 100`  
akkor: `ki (getSzék, getKm, getFt)`

<Utas> *utasok* :lista

**kiválogatás tétele ( getKm)**

**Utas** osztály

- szék :egészszám
- mettől :egészszám
- meddig :egészszám
- km :egészszám
- végig :logikai
- Ft :egészszám // 71 Ft/10km

+ getSzék

+ getKm

+ getFt

A jegyekből mennyi bevétele  
származott a társaságnak



<Utas> *utasok* :lista

összegzés tétele ( getFt)

Utas osztály

- szék :egészszám
- mettől :egészszám
- meddig :egészszám
- km :egészszám
- végig :logikai
- Ft :egészszám // 71 Ft/10km

+ getFt

A főprogramban végig az utasokon  
szum = szum + **getFt**

# Utas

- szék :egésszám
- mettől :egésszám
- meddig :egésszám
- km :egésszám
- végig :logikai
- Ft :egésszám // 71 Ft/10km

- + getSzék
- + getKm
- + isVégig
- + getFt
- + getMettől // + feladathoz
- + getMeddig // + feladathoz



# Konstruktor

```
public Utas(String sor) {  
    String[] tmp = sor.split(" ");  
    this.szék = Integer.parseInt(tmp[0]);  
    this.mettől = Integer.parseInt(tmp[1]);  
    this.meddig = Integer.parseInt(tmp[2]);  
    this.km = meddig - mettől;  
    setVégig();  
    setFt();  
}
```

- eljárás setFt()

$km10 = km / 10$  // egész osztás

ha  $(km \% 10 \neq 0)$  // maradékos osztás

akkor:  $km10 = km10 + 1;$

ha vége

//kerekítés

$ft = km10 * 71$

$maradék = ft \% 5$  // 0,1,2,3,4

ha  $(maradék > 2)$  // 3,4

akkor:  $ft = ft - maradék + 5$

különben:  $ft = ft - maradék$

ha vége

$km10 = 363 / 10 = 36$

$363 \% 10 = 3$

$km10 = 36 + 1 = 37$

$ft = 37 * 71 = 2627$

$maradék = 2$

$ft = 2627 - 2 + 5$

$ft = 2627 - 2$  2625