

ÚTÉPÍTÉS

Az **A** és **B** várost összekötő út 10000 m hosszú részének a felújításán dolgoznak. Ennek a szakasznak a forgalmát figyeljük egy nap néhány óráján (7 – 16)-keresztül. Az említett szakaszon előzési tilalom van érvényben.



A forgalmat a `forgalom.txt` állomány tartalmazza. Soronként egy áthaladó jármű adatai olvashatók időrendben. Egy sorban az első három szám azt az időpontot jelöli (óra, perc, másodperc), amikor a jármű belép a vizsgált útszakaszra. A következőszám jelöli, hogy a jármű az érintett távolságot hány másodperc alatt tenné meg – a belépéskor mért sebességgel –, ha haladását semmi nem akadályozná. Ezt egy betű követi, amely jelzi, hogy a jármű melyik város **irányából** érkezett. Ennek megfelelően a betű **A** vagy **B** lehet. Az egyes adatokat pontosan egy szóköz választja el egymástól.

-	óra	:int	
-	perc	:int	
-	mperc	:int	
-	sebesség	:double	// 10/(egésszám(temp[3])/3600.0) → km/h
-	irány	:String	
-	mp_idő	:int	//az óra, perc, mperc átváltása másodpercre
+	Adat	(String sor)	//konstruktor

1. Olvassa be a *forgalom.txt* állományban talált adatokat
objektumok beolvasása az **< Autó > autók** listába
2. Írja ki a képernyőre, hogy az n-edikként belépő jármű melyik város felé haladt! Ehhez kérje be a felhasználótól az n értékét!
lista n. elemének elérése
KI: *autók* lista n. elemének irány adattagja: *getIrány()*
3. Írja a képernyőre, hogy a **B** város irányába tartó utolsó két jármű hány másodperc különbséggel érte el az útszakasz kezdetét!
kiválasztás(n, k), lista n. és k. elemének elérése
a lista végéről előre, amíg *!(getIrány()↔"A")* d=lista(i).getMp_idő()
innét (i-1) tovább előre, amíg *!(getIrány()↔"A")* d= d-lista(i).getMp_idő()
ki: (d)
4. Statisztika: határozza meg óránként és irányonként, hogy hány jármű érte el a szakaszt! Soronként egy-egy óra adatait írja a képernyőre! Az első érték az órát, a második érték az **A**, a harmadik a **B** város felől érkező járművek számát jelentse! A kiírásban csak azokat az órákat jelenítse meg, amelyekben volt forgalom valamely irányban!
kiválogatás tömbbe
új A[10], B[10] //7,8,9,10,...,16,17
1.végig a listán:
 ha (getIrány() ↔ "A")
 akkor A[getOra()-7]++
 különben B[getOra()-7]++
2.végig az **A[]-n**: ki(i+7,". óra ",A[i], B[i])
5. A belépéskor mért értékek alapján határozza meg a 10 leggyorsabb járművet!
Írassa ki a képernyőre a várost (**A**, illetve **B**), amely felől érkezett, ezek belépési idejét oo:pp:ss formátumban, és km/h egységben kifejezett sebességét 1 tizedes pontossággal, sebességük szerinti csökkenő sorrendben!
Soronként egy jármű adatait jelenítse meg.

```

1. N = adatok.mérete();
   ciklus (i=0; i<N-1; i++)
       minIndex = i
       ciklus (j=i+1; j<N; j++)
           Ha (adatok.get(j).getSebesség() > adatok.get(minIndex).getSebesség())
               akkor minIndex = j;
           Ha vége
       ciklus vége
       Adat tmp = adatok.get(i) // lementem az i. elemet
       adatok.set(i, adatok.get(minIndex)) // felül írom az i. elemet
       adatok.set(minIndex, tmp) // felül írom a minIndex. elemet
   ciklus vége

```

rendezés A-Z *getSebesség()* szerint
2.első 10 elem kiírása a listából: