

1. feladat: Vírus (20 pont)

Biológusok egy különleges vírustörzset vizsgálnak. Egy kísérlet során minden egyedre feljegyezték annak keletkezési és pusztulási időpontját. Adott  $H$  értékre szeretnék kiszámítani, hogy melyik az a  $H$  hosszúságú időintervallum, amely alatt a legtöbb egyed volt megfigyelhető. A  $T$  kezdetű,  $H$  hosszú időintervallumban azok a  $K$  keletkezési és  $P$  pusztulási idejű egyedek voltak megfigyelhetők, amelyekre az alábbi feltételek teljesülnek:

**$T \leq K$  és  $K < T + H$ , vagy  $T < P$  és  $P < T + H$ , vagy  $K < T$  és  $P \geq T + H$ .**

Írj programot (VIRUS.PAS, VIRUS.C, ...), amely kiszámítja annak a  $H$  hosszú időintervallumnak a kezdetét, amely alatt a legtöbb egyed volt életben!

A VIRUS.BE szöveges állomány első az egyedek  $N$  ( $1 \leq N \leq 300000$ ) számát és az időintervallum  $H$  ( $1 < H \leq 7000$ ) értékét tartalmazza egy szóközzel elválasztva. A következő  $N$  sorban az egyes egyedek  $K$  keletkezési és  $P$  pusztulási ideje van, ( $1 \leq K < P \leq 7000$ ) egy szóközzel elválasztva.

A VIRUS.KI szöveges állomány első sorába két egész számot kell írni, az első a megfigyelhető vírusok maximális száma legyen, a második az a  $T$  időpont, amelyre a  $T$  kezdetű és  $H$  hosszú időintervallumban a legtöbb egyed volt megfigyelhető ( $T + H \leq 7000$ ). Több megoldás esetén a legkorábbi időpontot kell kiírni.

Példa:

VIRUS.BE

```
9 3
2 6
1 3
5 6
4 7
4 9
8 11
11 14
10 13
10 12
```

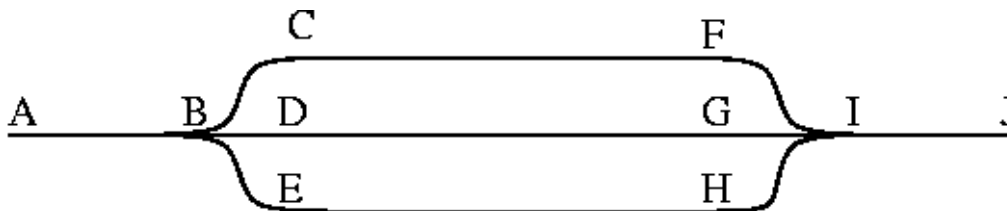
VIRUS.KI

```
4 2
```

2. feladat: Vasút (40 pont)

Háromsor város vasútállomásán nagy gondot okoz a szerelvények rendezése. Az állomásról továbbítandó szerelvényeket úgy kell kialakítani, hogy amikor az megérkezik a célállomásra, a szerelvény végéről mindig lekapcsolható legyen az oda továbbított kocsisor. Minden továbbítandó szerelvény négy állomást érint, ezért a rendezés előtt minden kocsit megjelölnek az 1, 2, 3 vagy 4 számokkal. A szerelvény kocsijait rendezzük át úgy, hogy a szerelvény elején legyenek az 1-essel, aztán a 2-essel, majd a 3-assal, végül a 4-essel megjelöltek. Kezdetben a kocsik az ábrán látható I-J pályaszakaszon vannak. A vasúti váltók működése csak a következő hat műveletet teszi lehetővé.

- B1 Az I-J szakaszból balról az első kocsit átmozgatni a C-F szakaszba a már ott lévő kocsik mögé.
- B2 Az I-J szakaszból balról az első kocsit átmozgatni a D-G szakaszba a már ott lévő kocsik mögé.
- B3 Az I-J szakaszban lévő első kocsit átmozgatni az E-H szakaszba a már ott lévő kocsik mögé.
- K1 Az C-F szakaszban lévő első kocsit átmozgatni és hozzáilleszteni az A-B szakaszon kialakítandó rendezett kocsisor végére.
- K2 Az D-G szakaszban lévő első kocsit átmozgatni és hozzáilleszteni az A-B szakaszon kialakítandó rendezett kocsisor végére.
- K3 Az E-H szakaszban lévő első kocsit átmozgatni és hozzáilleszteni az A-B szakaszon kialakítandó rendezett kocsisor végére.



Írj programot (VASUT.PAS, VASUT.C vagy VASUT.CPP néven), amely meghatároz egy olyan tolatási műveletsort, amelynek végrehajtása a bemeneti kocsisor rendezését eredményezi!

A VASUT.BE szöveges állomány első sora a kocsik  $N$  számát tartalmazza ( $1 \leq N \leq 1000$ ). A második sor pontosan  $N$  egész számot tartalmaz egy-egy szóközzel elválasztva, a rendezendő kocsik címkéit. Minden címke értéke 1, 2, 3 vagy 4.

A VASUT.KI szöveges állomány egy olyan tolatási műveletsort tartalmazzon, amelynek végrehajtása a bemeneti kocsisor rendezését eredményezi. Minden sor egy-egy tolatási művelet jelét tartalmazza, ami a 'B1', 'B2', 'B3', 'K1', 'K2' vagy 'K3' string lehet. Ha nem lehet rendezni a bemeneti kocsisort, akkor az állomány első és egyetlen sorába a NEM szót kell írni!

Példa:

VASUT.BE	VASUT.KI
5	B1
2 3 1 4 3	B3
	B2
	K2
	B1
	K1
	K3

B2  
K2  
K1

### 3. feladat: Találka (40 pont)

Rómeó és Júlia a lehető legrövidebb időn belül találkozni szeretne. Jelenleg egymástól távol, különböző városban vannak. Repülővel akarnak utazni egy olyan városba, ahova a legrövidebb idő alatt mindketten megérkezhetnek. Az útvonal kiválasztásához ismerik az összes igénybe vehető repülőjáratot.

Készíts programot (TALALKA.PAS, TALALKA.C, ...), amely a repülőjáratok, valamint Rómeó és Júlia tartózkodási helyének ismeretében megadja a legközelebbi találkozási pontot és azt a két útvonalat, amelyen közlekedniük kell ahhoz, hogy a lehető legkorábban találkozzanak!

A TALALKA.BE szöveges állomány első sorában a városok ( $1 \leq N \leq 100$ ) és a repülőjáratok ( $1 \leq M \leq 10000$ ) száma van, egyetlen szóközzel elválasztva. A városokat az  $1, \dots, N$  számokkal azonosítjuk. A második sorban a Rómeó és Júlia tartózkodási helyének sorszámja van ( $1 \leq A, B \leq N$ ). A következő  $M$  sor mindegyikében egy repülőjárat van ( $1 \leq K \neq V \leq N$ ), egyetlen szóközzel elválasztva. Ez azt jelenti, hogy a  $K$  városból van közvetlen egyirányú járat a  $V$  városba. Minden járat naponta csak egyszer közlekedik és reggel indul azonos időben.

A TALALKA.KI állomány három sort tartalmazzon. Az első sorba két számot kell írni, az első a legközelebbi találkozás ideje, a második a legközelebbi találkozási város sorszám legyen! Ha nincs ilyen, akkor az első sorba  $-1$ -et kell írni és ilyenkor a következő két sor legyen üres. A második sorba azt az útvonalat kell írni, amelyiken Rómeó eljut a találkozási városba, a harmadikba pedig azt az útvonalat, amelyiken Júlia eljut a találkozási városba. Ha több megoldás is van, egy tetszőlegeset ki lehet írni.

#### Példa:

TALALKA.BE

```
10 15
1 8
1 2
2 1
1 4
3 2
4 3
2 5
3 6
3 7
4 7
5 10
6 9
7 6
8 7
8 9
9 10
```

TALALKA.KI

```
2 7
1 4 7
8 7
```

