

## Mozgó szerelők

Egy vállalat különböző városokban lévő ügyfeleinek nyújt szolgáltatást. A vállalatnak három mozgó szerelője van. Ha kérés érkezik valamely helyről, – a kérés kielégítése érdekében – az egyik alkalmazottnak aktuális helyéről a kérés helyére kell utaznia (amennyiben még nincs ott alkalmazott). Bármely időpillanatban csak egy szerelő utazhat. Az alkalmazottak csak kérés esetén utazhatnak, és nem tartózkodhatnak egy időben egyazon helyen. Egy szerelő utazása a  $p$  helyről a  $q$  helyre adott  $C(p, q)$  költségbe kerül. A költségfüggvény nem szükségszerűen szimmetrikus, de a helyben maradás költsége 0, azaz  $C(p, p) = 0$ . A kéréseket a vállalatnak szigorúan beérkezési sorrendben kell kiszolgálnia. A cél egy adott kérésorozat összköltségének minimalizálása.

Írj programot, amely eldönti, hogy az egyes kérések kiszolgálásához mely alkalmazottnak kell utaznia, annak érdekében, hogy a megadott kérésorozat kiszolgálásának teljes költsége a lehető legkevesebb legyen!

### Bemenet

A *standard bemenet* első sorában a helyek száma ( $3 \leq L \leq 200$ ), és a kérések száma ( $1 \leq N \leq 1000$ ) van. A következő  $L$  sor mindegyike pontosan  $L$  nemnegatív költséget tartalmaz ( $1 \leq C(i, j) \leq 2000$ ). Az utolsó sor az  $N$  kérés helyét tartalmazza ( $1 \leq K_i \leq L$ ). Kezdetben a három alkalmazott rendre az 1, 2 és 3 helyeken tartózkodik.

### Kimenet

A *standard kimenet* első sorába a bemenő kérés-sorozat kiszolgálásához szükséges minimális összköltséget kell írni! A második sor  $i$ . száma annak az alkalmazottnak az azonosítója legyen (1, 2 vagy 3), aki az  $i$ . kérést kiszolgálja! Ha több megoldás is létezik, a programodnak elegendő egyet kiírnia, mindegy, hogy melyiket.

### Példa

Bemenet

```
5 9
0 1 1 1 1
1 0 2 3 2
1 1 0 4 1
2 1 5 0 1
4 2 3 4 0
4 2 4 1 5 4 3 2 1
```

Kimenet

```
5
1 2 1 2 2 1 3 1 3
```

### Korlátok

Időlimit: 0.5 mp.

Memórialimit: 64 MB