

## F. feladat: Ciklikus rendezés

Adott  $n$  darab pozitív egész szám tömbje  $a_1, a_2, \dots, a_n$ . A következő műveletet akárhányszor elvégezheted: válassz több különböző indexet  $i_1, i_2, \dots, i_k$  ( $1 \leq i_j \leq n$ ) és mozgasd az  $i_1$  helyen álló számot az  $i_2$  helyre, az  $i_2$  helyen álló számot az  $i_3$  helyre,  $\dots$ , az  $i_k$  helyen álló számot az  $i_1$  helyre. Más szóval, a művelet ciklikusan eltolja az elemeket:  $i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_k \rightarrow i_1$ . Az ilyen műveletet *ciklus*nak is nevezzük.

Például, ha van egy  $n = 4$  elemű tömb  $a_1 = 10, a_2 = 20, a_3 = 30, a_4 = 40$ , és három indexet választasz  $i_1 = 2, i_2 = 1, i_3 = 4$ , akkor az eredményül kapott tömb  $a_1 = 20, a_2 = 40, a_3 = 30, a_4 = 10$  lenne.

Írj programot, amely a tömböt nem csökkenő sorba rendezi minimális számú művelettel. További megszorítás, hogy az egyes műveletekben alkalmazott ciklusok hosszainak összege kisebb vagy egyenlő legyen  $s$ -nél. Ha nem lehetséges a tömböt lerendezni ezzel a megszorítással, akkor ezt is jelezned kell.

### Bemenet

A bemenet első sora két egész számot, a tömbben lévő elemek  $n$  számát és a ciklushosszak összegének felső  $s$  határát tartalmazza ( $1 \leq n \leq 200\,000$ ,  $0 \leq s \leq 200\,000$ ).

A következő sor  $n$  egészet tartalmaz  $a_1, a_2, \dots, a_n$ , a tömb elemeit ( $1 \leq a_i \leq 10^9$ ).

### Kimenet

Ha nem lehet a tömböt lerendezni úgy, hogy a ciklusok maximális hossza ne haladja meg  $s$ -et, írd ki egyetlen számot **-1**.

Különben írd ki egyetlen  $q$  számot, a tömb rendezéséhez szükséges műveletek minimális számát. A következő  $2 \cdot q$  sorban részletezd a műveleteket olyan sorrendben, ahogy a tömbön alkalmazni kell őket. Az  $i$ . művelet leírása egy olyan sorral kezdődik, amely egy  $k$  egészet tartalmaz ( $1 \leq k \leq n$ ), a ciklus hosszát (azaz a kiválasztott indexek számát). A következő sor  $k$  különböző egészet tartalmaz  $i_1, i_2, \dots, i_k$  ( $1 \leq i_j \leq n$ ), magukat az indexeket.

Ezen a ciklusok összhosszának kisebbnek vagy egyenlőnek kell lennie  $s$ -nél, és a tömbnek rendezettnek kell lennie a  $q$  művelet elvégzése után.

Ha több lehetséges megoldás is van az optimális  $q$ -val, bármelyiket kiírhatod.

## Pontozás

A feladat kilenc tesztcsoporthot tartalmaz, mindegyik tesztcsoporth pontszámát csak akkor kapod meg, ha a tesztcsoporthon belüli tesztek mindegyike sikeres.

1. (5 pont)  $n, s \leq 2$  és a tömb összes eleme vagy 1 vagy 2
2. (5 pont)  $n \leq 5$
3. (5 pont) A tömb összes eleme vagy 1 vagy 2
4. (10 pont) A tömb csak 1-től  $n$ -ig tartalmaz számokat, mindegyik pontosan egyszer szerepel,  $s = 2 \cdot n$
5. (10 pont) A tömb csak 1-től  $n$ -ig tartalmaz számokat, mindegyik pontosan egyszer szerepel,  $n \leq 1000$
6. (15 pont) A tömb csak 1-től  $n$ -ig tartalmaz számokat, mindegyik pontosan egyszer szerepel
7. (15 pont)  $s = 2 \cdot n$
8. (15 pont)  $n \leq 1000$
9. (20 pont) Nincs további megszorítás.

## Példák

### 1. példa

Bemenet:

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

Kimenet:

```
1
5
1 4 2 3 5
```

Ebben a példában le lehet rendezni a tömböt két olyan művelettel is, amelynek az összhossza 5: először alkalmazzuk az  $1 \rightarrow 4 \rightarrow 1$  (2 hosszú) ciklust, majd alkalmazzuk a  $2 \rightarrow 3 \rightarrow 5 \rightarrow 2$  (3 hosszú) ciklust. Ugyanakkor ez rossz válasz lenne, mert a műveletek minimális számát kérték, ami 1 ebben az esetben (ld. kimenet).

### 2. példa

Bemenet:

```
4 3
2 1 4 3
```

Kimenet:

```
-1
```

Ebben a példában le lehet rendezni a tömböt két olyan ciklussal is, amelynek az összhossza 4: ( $1 \rightarrow 2 \rightarrow 1$  és  $3 \rightarrow 4 \rightarrow 3$ ). Ugyanakkor ezt lehetetlen elérni rövidebb ciklusokkal, amit az  $s = 3$  követelt meg.

### 3. példa

Bemenet:

```
2 0  
2 2
```

Kimenet:

```
0
```

Ebben a példában a tömb már rendezett, nincs szükség további műveletekre. A nem létező ciklusok összhossza 0-nak tekinthető.

### 4. példa

Bemenet:

```
6 9  
6 5 4 3 2 1
```

Kimenet:

```
2  
6  
1 6 2 5 3 4  
3  
3 2 1
```

### 5. példa

Bemenet:

```
6 8  
6 5 4 3 2 1
```

Kimenet:

```
3  
2  
3 4  
4  
1 6 2 5  
2  
2 1
```

Figyeld meg, hogy az 1. és 3. példa ismétlődő számokat tartalmaz, így azok nem elégítik ki a 4., 5. és 6. tesztcsoporthoz követeleményeit. A 2., 4. és 5. példa megfelel az 5. és 6. tesztcsoporthoz követeleményeinek.