

# Dobozok

Az IOI2015 nyitóünnepségén minden csapat kap egy ajándékot egy dobozban, amit Amannak kell szétosztania.

Az ünnepség helyszíne  $L$  szektor kör alakban elrendezve,  $0$ -tól  $L - 1$ -ig sorszámozva, órajárással megegyező sorrendben. Azaz az  $i$ . szektort az  $i + 1$ . követi, ha  $0 \leq i \leq L - 2$ , és az  $L - 1$ -et a  $0$ . követi. Az  $N$  csapat mindegyike valamely szektorban ül, egy szektorban több is lehet, valamint lehet olyan szektor, ahol nem ül csapat.

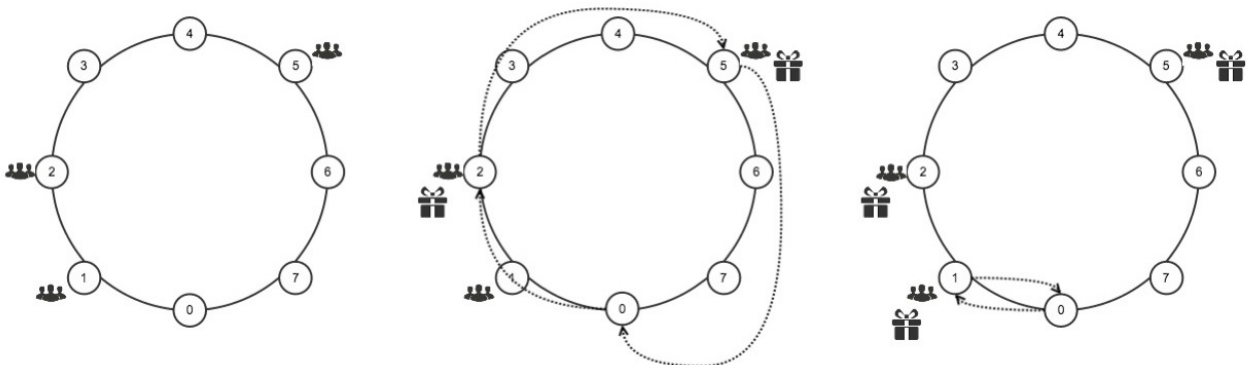
Kezdetben az  $N$  egyforma ajándék és Aman is a  $0$ . szektorban van. Aman minden csapatnak egy ajándékot visz és a végén vissza kell térnie a  $0$ . szektorba. Megjegyezzük, hogy a  $0$ . szektorban is lehet csapat.

Aman egyszerre legfeljebb  $K$  ajándékot vihet, amit a  $0$ . szektorban vesz fel. Aman mindkét irányban cirkulárisan haladhat, ha olyan szektohoz ér, ahol van még ajándékot nem kapott csapat, akkor annak adhat egyet. Csak a mozgásához van szüksége időre, egy szomszédos szektorba (mindkét irányban) egy másodperc alatt ér el.

Írj programot, amely kiszámítja, hogy legkevesebb hány másodperc alatt oszthatja ki az összes ajándékot!

## Példa

A példában  $N = 3$  csapat van, Aman egyszerre legfeljebb  $K = 2$  ajándékot vihet, a szektorok száma  $L = 8$ . A csapatok az 1., a 2., és az 5. szektorban vannak.



Az ábrán egy optimális megoldás látható. Aman először két ajándékot visz, ezeket a 2. és az 5. szektorban levő csapatnak adja, majd visszatér a 0. szektorba. Ehhez 8 másodperc kell. Másodjára Aman egy ajándékot visz az

1. szektorban levő csapatnak, majd visszatér a 0. szektorba, amihez 2 másodperc kell. Tehát az optimális megoldás értéke 10.

## Feladat

Adott  $N$ ,  $K$ ,  $L$  és a csapatok helye alapján számítsd ki, hogy minimálisan hány másodperc alatt osztható ki az összes ajándék! Megjegyzendő, hogy a végén is vissza kell térnie a 0. szektorba! A megoldáshoz a delivery függvényt kell megvalósítanod:

- `delivery(N, K, L, positions)` — ezt a függvényt az értékelő egyszer hívja.
  - $N$ : a csapatok száma.
  - $K$ : Aman által egyszerre vihető ajándékok maximális száma.
  - $L$ : a szektorok száma.
  - `positions`:  $N$  elemű tömb: `positions[0], ..., positions[N-1]`, ahol `positions[i]` az  $i$ . csapat sektorsorszáma. A `positions` tömb elemei nemcsökkenő sorrendben vannak.
  - A függvény visszaadott értéke a kiosztáshoz minimálisan szükséges másodpercek száma legyen!

## Részfeladatok

részfeladat	pont	$N$	$K$	$L$
1	10	$1 \leq N \leq 1,000$	$K = 1$	$1 \leq L \leq 10^9$
2	10	$1 \leq N \leq 1,000$	$K = N$	$1 \leq L \leq 10^9$
3	15	$1 \leq N \leq 10$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$
4	15	$1 \leq N \leq 1,000$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$
5	20	$1 \leq N \leq 10^6$	$1 \leq K \leq 3,000$	$1 \leq L \leq 10^9$
6	30	$1 \leq N \leq 10^7$	$1 \leq K \leq N$	$1 \leq L \leq 10^9$

## Megvalósítás

Egyetlen file-t kell beküldened: `volunteers.c`, `volunteers.cpp`, `volunteers.pas`, vagy `volunteers.java`, amiben a kívánt függvény van, az alábbi deklaráció szerint.

### C/C++ program (include `boxes.h` a file elején)

```
long long delivery(int N, int K, int L, int position[])
```

### Pascal programs (implement the described method in the

## **unit boxes)**

```
function delivery(N, K, L: longint, position: array of longint): int64
```

## **Java programs (implement the described method in the public class boxes)**

```
public long delivery(int N, int K, int L, int[] position)
```

## **Minta értékelő**

Az alábbiak szerint olvassa a bemenetet:

- Az 1. sor: N K L
- A 2. sor: positions[0] ... positions[N-1]

Kiírja a delivery függvény értékét.