



### 3. Feladat – order

100 pont

Tekintsük az összes véges, nem nulla természetes számokból álló számsorozatokot ekképpen rendezve:

[1]; [1,1]; [2]; [1,1,1]; [1,2]; [2,1]; [3]; [1,1,1,1]; [1,1,2]; [1,2,1]; [1,3]; ...

A rendezés a következő szabály szerint történik: ha van két különböző elemösszegű sorozatunk, akkor a kisebb elemösszegű sorozat található a kisebbik pozícióban. Ha két egyenlő elemösszegű sorozatunk van, akkor összehasonlítjuk a megfelelő pozíciókon levő tagokat a sorozatokban, ameddig találunk két különböző elemet. Az a sorozat amelyik a kisebb elemet tartalmazza, kisebb pozíción található.

Más szóval, az első rendezési kritérium a tagok összege, és ha a tagok összege egyenlő, a második rendezési kritérium a lexikografikus sorrend. Mindenik sorozathoz hozzárendelünk egy pozíciót (nemnulla természetes szám) és fordítva, mindenik pozícióhoz hozzárendelünk egy sorozatot.

**Példa:** Az [1,1,2] sorozathoz hozzárendeljük a 9-es pozíciót.

A 14-es pozícióhoz hozzárendeljük a [3,1] sorozatot.

#### Követelmény

Válaszolatok egy sor lekérdezésre melyek típusai a következők lehetnek:

1. Ismerve egy nem nulla természetes számokból álló sorozatot határozzátok meg sorozathoz hozzárendelt pozíciót.
2. Ismerve egy természetes számot amely egy pozíciót jelöl, határozzátok meg a megfelelő sorozatot.

#### Bemeneti adatok

Az **order.in** bemeneti állomány első sora tartalmazza a **Q** természetes számot, amely a lekérdezések számát jelenti.

A következő **Q** sor tartalmazza a lekérdezéseket.

Ha a lekérdezés 1-es típusú, akkor a sor tartalmazza az 1-es számot, majd egy **N** természetes számot, amely a sorozat tagjainak számát jelenti, majd **N** darab természetes számot amelyek a sorozat tagjait jelentik.

Ha a lekérdezés 2-es típusú, akkor a sor tartalmazza az 2-es számot, majd egy **P** nemnulla természetes számot, amely a keresett sorozat pozícióját jelenti.

A lekérdezéseket tartalmazó sorokban a számok egyetlen szóközzel vannak elválasztva egymástól.

#### Kimeneti adatok

Az **order.out** kimeneti állomány **Q** sort kell tartalmazzon. Mindenik sor tartalmazza a bemeneti állomány megfelelő lekérdezésére adott választ.

Ha a lekérdezés 1-es típusú akkor a megfelelő sorba egyetlen **P** számot írunk, ami a lekérdezésben megadott sorozat pozícióját jelenti.

Ha a lekérdezés 2-es típusú akkor a megfelelő sorba egy **N** természetes számot írunk, ami a kért sorozat elemeinek számát jelenti, majd **N** darab nemnulla természetes számot, amelyek a sorozat tagjait jelentik. Ezekeben a sorokban a számok egyetlen szóközzel kell legyenek elválasztva.

#### Megkötések és pontosítások

- $1 \leq P \leq 10^{15}$  (pontosabban biztosított, hogy mindkét típusú lekérdezésben a sorozathoz hozzárendelt pozíció értéke nem haladja meg  $10^{15}$  értékét)
- $1 \leq Q \leq 10^5$
- 40 pont értékben a tesztek csak 1-es típusú lekérdezéseket tartalmaznak
- 40 pont értékben a tesztek csak 2-es típusú lekérdezéseket tartalmaznak
- 20 pont értékben a tesztek mindkét típusú lekérdezést tartalmazzák

**Példa:**

order.in	order.out	Magyarázatok
2	9	A bemeneti állomány két típusú lekérdezést tartalmaz.
1 3 1 1 2	2 3 1	Az első lekérdezés 1-es típusú és a 3 hosszúságú [1,1,2] sorozat pozíciójának meghatározását kéri. Ennek a sorozatnak a pozíciója 9, a leírt sorrend alapján.
2 14		A második lekérdezés a 14. pozíción található sorozat meghatározását kéri. Ez pedig a 2 hosszúságú [3,1] sorozat.

**Maximális végrehajtási idő/teszt: 1, 2 sec**

**Rendelkezésre álló össz memória: 64 MB**

**A forráskód maximális mérete: 20 KB**