# Orvos

Betegek érkeznek egy orvoshoz különböző időpontokban, kezelésük érkezés szerinti sorrendben történik. Tudjuk mindenkiről, hogy hány percig tart a kezelése.

Készíts programot, amely megadja, hogy ki várakozott a legtöbbet a kezelése megkezdése előtt és maximum hányan várakoztak egyszerre!

## Bemenet

A *standard bemenet* első sorában a betegek száma van (2≤N≤100000). A második sorban a betegek érkezési időpontja (1≤Erki≤10000000) és kezelési időtartama (1≤Ki≤100) található, érkezés szerinti sorrendben. Ha többen érkeznek ugyanabban az időpontban, akkor az orvos előbb a kisebb sorszámúval foglalkozik.

## Kimenet

A *standard kimenet* első sorába a legtöbbet várakozó sorszámát és várakozási idejét kell írni, több megoldás esetén a legkisebb sorszámút! A második sorba a várakozók maximális száma kerüljön! Ha senkinek nem kell várni, akkor mind a három kiírt szám 0 legyen!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 6 1 4 1 2 5 6 6 4 6 6 20 3 | 5 11 4 |

## Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Pontozás

A pontszám 50%-a szerezhető olyan tesztekkel, ahol N≤100 és Erki≤1000.

# Többszörös összefésülés

Ismerjük egy iskola osztályainak névsorát, ábécésorrendben. Az összefésülés algoritmus két rendezett sorozatot tud összefésülni egyetlen rendezett sorozattá, melynek lépésszáma a két rendezett sorozat elemszámának összege.

Készíts programot, amely megadja, hogy minimum hány lépésben lehet összefésülésekkel megkapni az iskola névsorát ábécésorrendben!

## Bemenet

A *standard bemenet* első sorában az osztályok száma van (1≤N≤100000). A következő sorban egy-egy osztály tanulói száma található (1≤Ti≤1000).

## Kimenet

A *standard kimenet* első sorába a minimális lépésszámot kell írni, amivel összefésülésekkel megkapható az iskola rendezett névsora!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 3 10 3 4 | 24  Magyarázat: először összefésüljük a második és harmadik osztályt, lépésszáma 7, utána ezzel összefésüljük az első osztályt, ennek lépésszáma 17, azaz összesen 24 lépést kell megtennünk. |

## Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

# Legnagyobb szintkülönbség

Egy hegymászó útja tervezésekor N helyen kérdezte le a felszín tengerszint feletti magasságát. Ha az i. helyen áll, akkor a hátralevő úton a megteendő legnagyobb szintkülönbség növekedés annál a j>i helynél van, ahol a j. és az i. pontban mért magasság különbsége a lehető legnagyobb.

Készíts programot, amely megad két helyet, amelyek között a legnagyobb szintkülönbség növekedés a lehető legnagyobb!

## Bemenet

A *standard bemenet* első sorában a helyek száma van (2≤N≤100000). A második sorban a tengerszint feletti magasságok találhatók (1≤Magi≤10000000).

## Kimenet

A *standard kimenet* első sorába a lehető legnagyobb szintkülönbség értékét és kettő, ehhez a szintkülönbséghez tartozó hely sorszámát kell írni (több megoldás esetén bármelyiket)! Ha a magasságok sehol nem nőnek, akkor egyetlen -1-et kell kiírni!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 10 9 3 5 4 8 1 2 3 4 1 | 5 2 5  Magyarázat: 5 szintkülönbséget kell megtenni felfelé a második és az ötödik hely között. |

## Korlátok

Időlimit: 0.1 mp.

Memórialimit: 32 MB

## Pontozás

A pontszám 50%-a szerezhető olyan tesztekkel, ahol N≤100.