

2-es feladat: ramen

100 pont

Nemrég nyitottál egy japán-vendéglőt, és nem megy minden fényesen. Néha a vendégek nagyon sokat várnak a rendelt ételre, és te azt hiszed, hogy végre megértetted ennek az okát.

A vendéglőnek nincsenek asztalai, egyetlen bárpultja van, amely egy futószalaggal rendelkezik, ezen jutnak el az ételadagok a konyhából a vendégekhez. A bárban van **500.000.000** szék növekvően sorszámozva 1-től kezdve a konyha irányából. Időnként a vendégek új rendeléseket tesznek le. Egy rendelés a **T**-ik másodpercben a **P**-ik széken ülő vendég által azonnal a konyhába jut. Az étel elkészítése **D** másodpercbe kerül, majd felrakják a futószalagra és pontosan **P** másodperc alatt eljut a vendéghez. Ez idő alatt az ételadag elvonul **1, 2, ... P - 1** sorszámú székek előtt. Ha valamilyen okból kifolyólag a vendég nem emeli le a rendelt ételadagját a futószalagról, akkor ez tovább folytatja útját, ellenkező esetben az étel **T + D + P** másodperc alatt jut el a rendelőjéhez.

Pillanatnyilag a vendéglő egyetlen fajta ételt, *ramen* szolgál fel. Ezáltal a vendégek nagyon könnyen összecserélhetik adagjaikat, ezt a lehetőséget szívesen ki is használják.

Adva vannak a következők:

- Egy vendégnek lehet nulla rendelése, vagy esetleg több rendelése is.
- Egy vendég amelyiknek nulla rendelése van, teljesen inaktív.
- Ha egy vendég a **T**-ik másodpercben rendel, akkor a függőben lévő rendelések száma pontosan a **T**-ik másodpercben fog nőni 1-gyel.
- Ha egy vendégnek van egy függőben levő rendelése, akkor le fogja emelni az első adag ramen a futószalagról, amelyik elvonul előtte, attól függetlenül, hogy ő rendelte vagy sem. Ha ezt megtette a **T**-ik másodpercben, akkor a függőben levő rendelések száma eggyel csökkenni fog pontosan a **T**-ik másodpercben.

Példaképpen vizsgáljuk meg a helyzetet a következő két rendelés esetén:

A ramen elkészítési ideje **D = 2** másodperc. Ez az idő konstans, és minden rendelés esetén figyelembe kell venni.

A **T1 = 10**-ik másodpercben a **P1 = 8**-as széken ülő személy (nevezzük **A**-nak) rendel egy adag ramen. A **T1 + D = 12**-ik másodpercben futószalagra kerül az étele.

A **T2 = 16**-ik másodpercben a **P2 = 6**-os széken ülő személy (nevezzük **B**-nek) rendel egy adag ramen. A **T2 + D = 18**-ik másodpercben futószalagra kerül az étele.

A **19**-ik másodpercben az **A**-nak szánt ételadag elmegy a **6**-os számú szék előtt, a **B** vendég, aki ugyancsak az ételadagjára vár, leveszi a futószalagról, és megeszi. Tehát a **19**-ik másodpercben enni fog, majd az általa rendelt ételadagot figyelmen kívül fogja hagyni, amikor ez el fog menni mellette a futószalagon.

A **28**-ik másodpercben az **B**-nek szánt ételadag eljut **A**-hoz, aki elveszi és megeszi a **28**-ik másodpercben.

Megállapíthatjuk, hogy általában véve igaz, hogy a késéseket leszámítva, minden vendég pontosan annyi adagot fog enni amennyit rendelt.

Követelmények

Hogy fel tudd mérni ennek a szokásnak a következményét, megszerezted az adatokat a mai nap összes rendeléseiről. Meg szeretnéd tudni minden rendelés esetén a következő értéket: ha egy rendelés egy adott vendég **NR**-ik rendelése, akkor a hányadik másodpercben fogja az illető vendég a **NR**-ik adagját megenni?

Bemeneti adatok

A `ramen.in` bemeneti állomány első sora tartalmazza a rendelések **N** számát, illetve a ramen **D** elkészítési idejét. A következő **N** sor mindegyike két számot fog tartalmazni: **T[i]** az **i**-edik rendelés ideje, illetve **P[i]**, a szék pozíciója, ahonnan az **i**-edik rendelés érkezett. Biztosítva van, hogy a rendelési idők páronként különbözőek, és növekvő sorrendben vannak a beolvasási sorrend szerint.

Kimeneti adatok

A `ramen.out` kimeneti állomány N sort fog tartalmazni, mindegyik sor egyetlen számot: az i -edik sor az i -edik rendelésre adott választ fogja tartalmazni a beolvasott adatok sorrendjében.

Megkötések és pontosítások

1. $1 \leq N \leq 100.000$
2. $0 \leq D, T[i] \leq 500.000.000$, minden $1 \leq i \leq N$ esetén.
3. $1 \leq P[i] \leq 500.000.000$, minden $1 \leq i \leq N$ esetén.
4. Biztosított, hogy $T[i] < T[i + 1]$, minden $1 \leq i < N$ esetén.
5. **22** pont megszerzése érdekében, a bemeneti adatoknál biztosítva van az általános megkötések mellett, hogy $N \leq 2000$ és $D, T[i], P[i] \leq 5000$
6. Más **25** pont megszerzése érdekében, a bemeneti adatoknál biztosítva van az általános megkötések mellett, hogy $N \leq 2000$.

Példák

ramen.in	ramen.out	Explicații și comentarii
2 2 10 8 16 6	28 19	A kijelentésben megadott példa
3 2 5 4 6 4 7 3	12 13 10	Láthatjuk, hogy a 4-es széken ülő vendég 2 rendelést tesz le. Az első válasz az a másodperc amikor az első rendelést leadó vendég először eszik. A második válasz az az időpont amikor a második rendelésért letevő vendég másodszor enni fog.
3 0 0 6 3 3 4 5	10 3 8	Láthatjuk, hogy a 3-as széken ülő vendég a 3-ik másodpercben rendel. Ugyanabban a másodpercben megjelenik előtte egy ramen és ő azonnal elfogyasztja.

Maximális futási idő: 1,8 sec

Memória: 128 MB amiből a verem 32 MB

A forráskód maximális mérete: 20 KB