Előzés (Overtaking)

A Budapesti Repülőtér és a Forrás Hotel között egy egysávos, egyirányú út van. Az út hossza L kilométer.

Az IOI2023 rendezvény ideje alatt $N+1$ transzferbusz közlekedik ezen az úton. A buszok számozása 0-tól N-ig tart. Az i. busz ($0 \le i < N$) a tervek szerint a rendezvény $T[i]$-edik másodpercében indul a repülőtérről, és $W[i]$ másodperc alatt képes 1 kilométert megtenni. Az N. busz tartalék, amely X másodperc alatt képes 1 kilométert megtenni. Az Y időpont, amikor ez a busz elindul a repülőtérről, még nem dőlt el.

Az úton általában tilos az előzés, de az **előzési pontokon** a buszok megelőzhetik egymást. Az út különböző helyein M darab ($M > 1$) előzési pont van, amelyeket 0-tól $M-1$-ig számoztak. A j. előzési pont ($0 \le j < M$) a repülőtértől az út mentén $S[j]$ kilométerre található. Az előzési pontok a repülőtértől kezdve növekvő sorrendbe vannak rendezve, azaz $S[j] < S[j+1]$ minden $0 \le j \le M-2$ esetén. Az első előzési pont a repülőtér, az utolsó pedig a szálloda, azaz $S[0] = 0$ és $S[M-1] = L$.

Minden busz a saját maximális sebességével halad, kivéve, ha utolér egy lassabb buszt, amelyik előtte halad az úton. Ekkor a buszok konvojban mennek tovább, és kénytelenek ugyanazzal a kisebbik sebességgel haladni, amíg el nem érik a következő előzési pontot. Az előzési pontokon a gyorsabb buszok megelőzik a lassabb buszokat.

Formálisan minden i és j esetén, ahol $0 \le i \le N$ és $0 \le j < M$, a $t_{i,j}$ időt (másodpercben), amikor az i busz **megérkezik** a j. előzési pontra, a következőképpen határozzuk meg. Legyen $t_{i,0} = T[i]$ minden $0 \le i < N$ esetében, és legyen $t_{N,0} = Y$. Egyébként minden olyan j esetén, ahol $0 < j < M$:

- Definiáljuk az i. busz j előzési pontra való **várható érkezési idejét** (másodpercben) annak az időnek, amikor az i. busz megérkezne a j. előzési pontra, ha teljes sebességgel haladna a $j-1$. előzési pontra való érkezésétől számítva. Vagyis legyen
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ minden $0 \le i < N$ esetén, és
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Az i. busz a j. előzési pontra az i. busz és minden olyan busz várható érkezési idejének *maximumán* érkezik, amely korábban érkezett a $j-1$. állomásra, mint az i. busz. Formálisan legyen $t_{i,j}$ az $e_{i,j}$ és minden olyan $e_{k,j}$ maximuma, amelyre $0 \le k \le N$ és $t_{k,j-1} < t_{i,j-1}$.

Az IOI szervezői a tartalék buszt akarják ütemezni (N. busz). A feladatod, hogy válaszolj a szervezők Q darab kérdésére, amelyek a következő formájúak: Adott az Y időpont (másodpercben), amikor a tartalék. busznak el kellene indulnia a repülőtérről. Ez a busz mikor érkezne meg a szállodába?

Megvalósítás

A következő függvényeket kell megvalósítanod:

```
void init(int L, int N, int64[] T, int[] W, int X, int M, int[] S)
```

- L: az út hossza.
- N: a menetrendjük szerinti közlekedő buszok száma.
- T: egy N hosszúságú tömb, amely leírja, hogy a nem tartalék buszok mikor indulnak a repülőtérről.
- W: egy N hosszúságú tömb, amely a nem tartalék buszok maximális sebességét adja meg.
- X: az az idő, amely alatt a tartalék busz 1 kilométert tud megtenni.
- M: az előzési pontok száma.
- S: egy M hosszúságú tömb, amely az előzési pontok távolságát írja le a repülőtértől.
- Ezt az eljárást minden tesztesetnél pontosan egyszer hívjuk meg, az `arrival_time` hívása előtt.

```
int64 arrival_time(int64 Y)
```

- Y: az az időpont, amikor a tartalékbusznak el kell indulnia a repülőtérről.
- Ennek az eljárásnak azt az időpontot kell visszaadnia, amikor az N. busz megérkezik a szállodába.
- Ezt az eljárást pontosan Q alkalommal hívjuk meg.

Példa

Nézzük a következő függvényhívásokat:

```
init(6, 4, [20, 10, 40, 0], [5, 20, 20, 30], 10, 4, [0, 1, 3, 6])
```

Figyelmen kívül hagyva a 4. tartalék buszt, a következő táblázat a többi busz várható és tényleges érkezési idejét mutatja az egyes előzési pontokon:

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180

A 0. előzési ponton az érkezési időpontok azok az időpontok, amikor a buszok menetrend szerint elhagyják a repülőteret. Vagyis $t_{i,0} = T[i]$ az $0 \le i \le 3$ esetén.

Az 1. előzési pontra való érkezés várható és tényleges időpontjait a következőképpen számítjuk ki:

- Az érkezések várható ideje az 1. előzési pontonon:
    - 0. busz: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
    - 1. busz: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
    - 2. busz: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
    - 3. busz: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Az 1. előzési pontra érkezés időpontjai:
    - Az 1. és a 3. buszok korábban érkeznek a 0. előzési pontra, mint a 0. busz, így $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
    - A 3. busz korábban érkezik a 0. előzési pontra, mint az 1. busz, így $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
    - A 0. busz, az 1. busz és a 3. busz a 0. előzési pontra korábban érkezik, mint a 2. busz, így $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
    - Egyetlen busz sem érkezik a 0. előzési pontra korábban, mint a 3 busz, így $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

A 4. busznak 10 másodpercbe telik 1 kilométer megtétele, és a menetrend szerint a 0. másodpercben indul a repülőtérről. Ebben az esetben a következő táblázat az egyes nem-tartalék buszok várható és valóságos érkezési idejét mutatja. Az eredeti táblázathoz képest bekövetkezett változások aláhúzva vannak.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	$\underline{60}$
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		0		10	10		30	30		60	60

Látjuk, hogy a 4. busz a 60. másodpercben érkezik a szállodához. A függvénynek tehát 60-t kell visszaadnia.

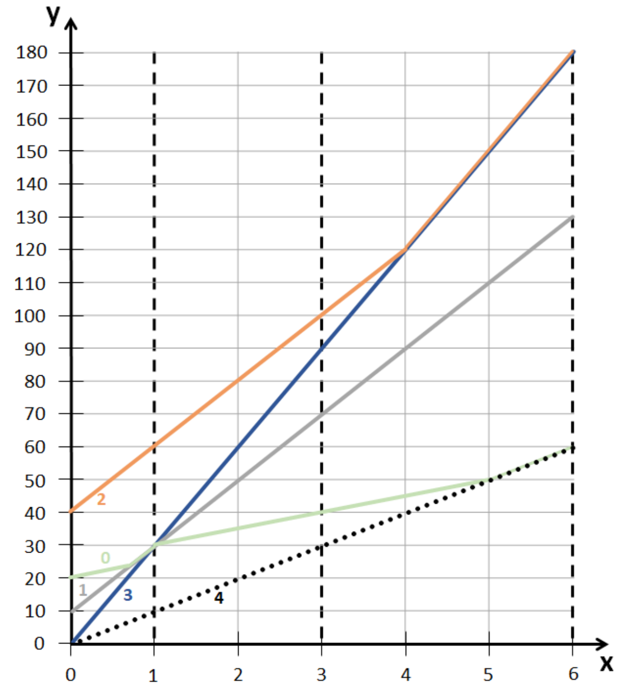
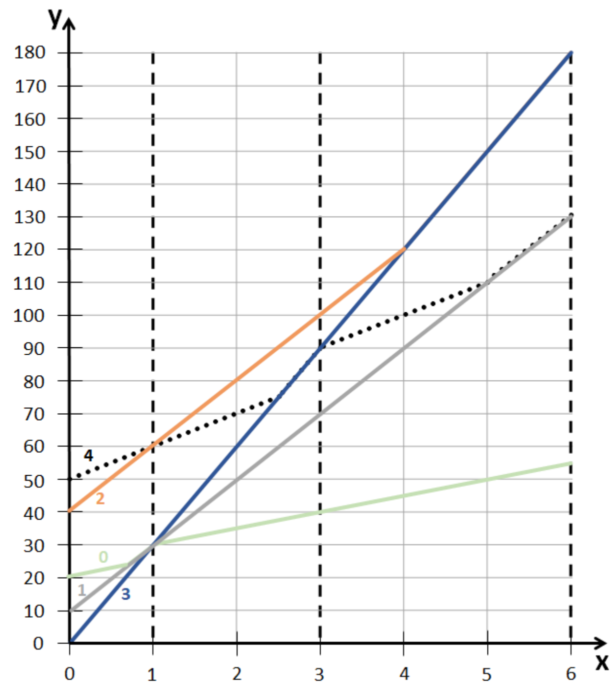
```
arrival_time(50)
```

A 4. busz most a tervek szerint az 50. másodpercben indul a repülőtérről. Ebben az esetben a nem-tartalék buszok érkezési ideje nem változik az eredeti táblázathoz képest. Az érkezési időpontokat a következő táblázat mutatja.

i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		50		60	60		80	90		120	130

A 4. busz megelőzi a lassabb 2. buszt az 1 előzési ponton, mivel egy időben érkeznek. Ezután a 4. busz az 1. és a 2. állomás között a 3. buszt utoléri, így a 4. busz a 80. helyett a 90. másodpercben érkezik a 2. állomásra. A 2. állomás elhagyása után a 4. busz az 1. busszal együtt érkezik a szállodához. A 4. busz a 130. másodpercben érkezik a szállodához. Az eljárásnak tehát 130-as eredményt kell adnia.

Ábrázolhatjuk a buszok a reptérről minden egyes távolságra történő megérkezéséhez szükséges időt. A diagram x-tengelye a repülőtértől való távolságot jelöli (kilométerben), a diagram y-tengelye pedig az időt (másodpercben). A függőleges szaggatott vonalak az előzési pontok helyzetét jelölik. A különböző folytonos vonalak (a buszindexek kíséretében) a négy menetrend szerinti, nem tartalék autóbuszt jelölik. A szaggatott fekete vonal a tartalék buszt jelöli.

`arrival_time(0)`	`arrival_time(50)`
	

Feltételek

- $1 \le L \le 10^9$
- $1 \le N \le 1\,000$
- $0 \le T[i] \le 10^{18}$ (minden i-re, ahol $0 \le i < N$)
- $1 \le W[i] \le 10^9$ (minden i-re, ahol $0 \le i < N$)
- $1 \le X \le 10^9$
- $2 \le M \le 1\,000$
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$
- $1 \le Q \le 10^6$
- $0 \le Y \le 10^{18}$

Részfeladatok

1. (9 pont) $N = 1, Q \le 1\,000$
2. (10 pont) $M = 2, Q \le 1\,000$
3. (20 pont) $N, M, Q \le 100$
4. (26 pont) $Q \le 5\,000$
5. (35 pont) Nincs további megkötés.

Mintaértékelő

A mintaértékelő a következő formátumban olvassa be a bemenetet:

- 1. sor: $L \ N \ X \ M \ Q$
- 2. sor: $T[0] \ T[1] \ \ldots \ T[N-1]$
- 3. sor: $W[0] \ W[1] \ \ldots \ W[N-1]$
- 4. sor: $S[0] \ S[1] \ \ldots \ S[M-1]$
- $5+k$. sor ($0 \le k < Q$): Y a k. kérdésre

A mintaértékelő a következő formátumban írja ki a menetet:

- $1+k$. sor ($0 \le k < Q$): az `arrival_time` függvény visszatérési értéke a k. kérdésre