Overtaking

Er bestaat een eenbaansweg in slechts één richting van Budapest Airport naar Hotel Forrás. De weg is L kilometer lang.

Tijdens de IOI 2023 rijden er $N+1$ bussen over deze weg. De bussen zijn genummerd van 0 tot en met N. Bus i ($0 \le i < N$) is gepland om te vertrekken op de $T[i]$-de seconde van de IOI en kan 1 kilometer rijden in $W[i]$ seconden. Bus N is een reservebus en kan 1 kilometer rijden in X seconden. Het tijdstip Y wanneer de reservebus het vliegveld verlaat, ligt vooraf niet vast.

Inhalen op de weg is niet toegestaan, maar bij de **passeerstations** mag dat wel. Er zijn M $(M > 1)$ passeerstations, genummerd van 0 tot en met $M-1$. De passeerstations j ($0 \le j < M$) bevinden zich op $S[j]$ kilometer van het vliegveld langs de weg. De passeerstations zijn gesorteerd in oplopende volgorde van afstand vanaf het vliegveld, met $S[j] < S[j+1]$ voor elke $0 \le j \le M-2$. Het eerste passeerstation is bij het vliegveld en de laatste is bij het hotel, dus $S[0] = 0$ en $S[M-1] = L$.

Elke bus rijdt op de maximum snelheid, behalve als het wordt opgehouden door een langzamere bus voor hem, in dat geval wordt de bus gedwongen om ook op een lagere snelheid te rijden totdat het volgende passeerstation wordt bereikt. Daar halen de snellere bussen de langzamere bussen in.

Netjes gezegd, voor elke i en j, met $0 \le i \le N$ and $0 \le j < M$, is het tijdstip $t_{i,j}$ (in seconden) wanneer bus i **aankomt bij** passeerstation j als volgt gedefinieerd. Voor $j = 0$ is $t_{i,0} = T[i]$ voor elke $i < N$ en is $t_{N,0} = Y$. Voor de andere gevallen voor j met $0 < j < M$ geldt:

- Definieer het *verwachte tijdstip van aankomst* (in seconden) van bus i bij passeerstation j, genoteerd als $e_{i,j}$, als het tijdstip wanneer bus i zou arriveren bij passeerstation j wanneer het zou rijden op volle snelheid vanaf het tijdstip het zou arriveren bij passeerstation $j-1$. In formulevorm:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ voor elke $i < N$, en
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Bus i arriveert bij passeerstation j op het maximum van de verwachte aankomsttijden van bus i en van iedere andere bus die aankomt bij station $j-1$ eerder dan bus i. Netjes gezegd, laat $t_{i,j}$ het maximum zijn van $e_{i,j}$ en iedere $e_{k,j}$ waarvoor $0 \le k \le N$ en $t_{k,j-1} < t_{i,j-1}$.

De IOI-organisatie wil de reservebus inplannen (bus N). Het is jouw taak om Q vragen te beantwoorden van de organisatoren, welke van de volgende vorm zijn: gegeven het tijdstip Y (in seconden) wanneer de reservebus zou moeten vertrekken vanaf het vliegveld, op welk tijdstip zou die bus dan arriveren bij het hotel?

Implementatiedetails

Het is jouw taak om de volgende functies te implementeren.

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

- L: de lengte van de weg.
- N: het aantal geplande bussen.
- T: een array van grootte N dat de vertrektijden bevat wanneer de gewone bussen gepland zijn om te vertrekken vanaf het vliegveld.
- W: een array van grootte N dat de maximum snelheden bevat van de gewone bussen.
- X: de tijd die nodig is om de reservebus 1 kilometer te laten rijden.
- M: het aantal passeerstations.
- S: een array van grootte M dat de afstanden van de passeerstations bevat vanaf het vliegveld.
- Deze functie wordt precies eenmaal aangeroepen voor iedere testcase, voordat er aanroepen moeten worden gedaan van `arrival_time`.

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

- Y: het tijdstip waarop de reservebus (bus N) zou moeten vertrekken vanaf het vliegveld.
- Deze functie moet het tijdstip retourneren dat de reservebus zou aankomen bij het hotel.
- Deze functie wordt precies Q keer aangeroepen.

Voorbeeld

Beschouw de volgende reeks van aanroepen:

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

Afgezien van bus 4 (die nog niet is ingepland), laat de volgende tabel de verwachte en de werkelijke aankomsttijden zien van de gewone bussen bij ieder passeerstation:

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

De tijdstippen van aankomst bij station 0 zijn de tijdstippen op welke de bussen gepland zijn te vertrekken bij het vliegveld. Dus, $t_{i,0} = T[i]$ voor $i = 0, 1, 2$ and 3.

De verwachte en werkelijke aankomsttijden bij passeerstation 1 zijn als volgt berekend:

- De verwachte aankomsttijden bij station 1:
  - Bus 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Bus 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Bus 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Bus 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- De werkelijke aankomsttijden bij station 1:
  - De bussen 1 and 3 arriveren bij station 0 eerder dan bus 0, dus $t_{0,1} = \max(e_{0,1}, e_{1,1}, e_{3,1}) = 30$.
  - Bus 3 arriveert bij station 0 eerder dan bus 1, dus $t_{1,1} = \max(e_{1,1}, e_{3,1}) = 30$.
  - Bus 0, bus 1 en bus 3 arriveren bij passeerstation 0 eerder dan bus 2, dus $t_{2,1} = \max(e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}) = 60$.
  - Geen enkele bus arriveert eerder bij station 0 dan bus 3, dus $t_{3,1} = \max(e_{3,1}) = 30$.

```
arrival_time(0)
```

Bus 4 doet 10 seconden over 1 kilometer en is nu gepland om vanaf het vliegveld te vertrekken bij de 0-de seconde. In dit geval, laat de volgende tabel de aankomsttijden zien van iedere bus. De veranderingen vergeleken met de eerste tabel zijn onderstreept.

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

We zien dat bus 4 aankomt bij het hotel in de 60-ste seconde dus moet de functie 60 retourneren.

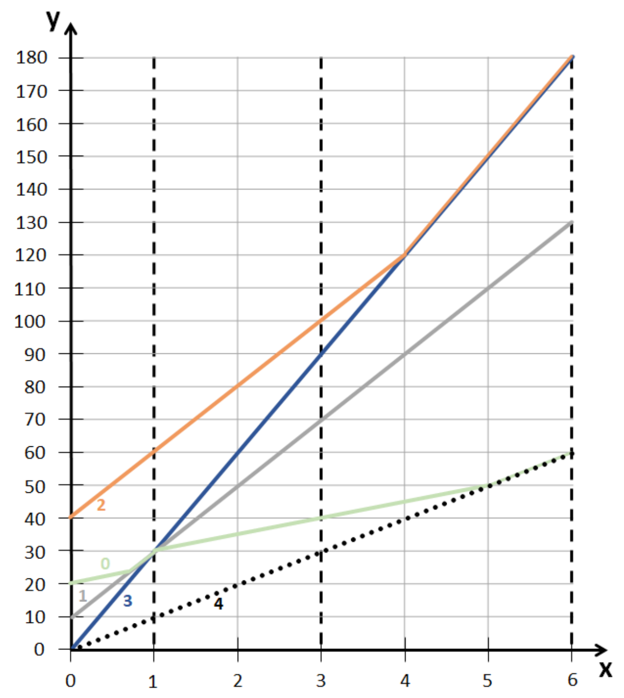
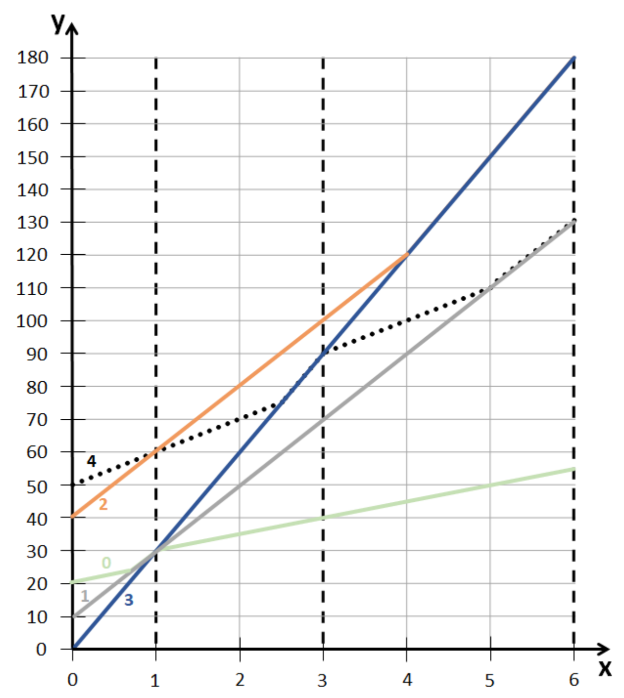
```
arrival_time(50)
```

Bus 4 is nu gepland om te vertrekken vanaf het vliegveld bij de 50-ste seconde. In dit geval zijn er geen veranderingen in de aankomsttijden voor de geplande bussen. De aankomsttijden zijn zichtbaar in de volgende tabel.

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

Echter haalt bus 4 de langzamere bus 2 in bij passeerstation 1 en arriveert op hetzelfde tijdstip. Daarna vertraagt bus 4 door bus 3 tussen de stations 1 en 2, zodat bus 4 aankomt op station 2 bij de 90-ste seconde in plaats van de 80-ste seconde. Na het vertrekken bij station 2 vertraagt bus 4 achter bus 1 totdat ze samen aankomen bij het hotel. Bus 4 komt aan bij het hotel in de 130-ste seconde dus moet de functie 130 retourneren.

In de volgende figuren zie je de afstanden (in kilometers) die de bussen hebben afgelegd vanaf het vliegveld tegen de tijd (in seconden). De verticale gestreepte lijnen geven de posities van de passeerstations aan. De gekleurde lijnen (voorzien van de bus-indices) geven de vier gewone bussen aan. De gestippelde zwarte lijn stelt de reservebus voor.

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

Randvoorwaarden

- $1 \leq L \leq 10^9$

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

Subtaken

1. (9 punten) $N = 1, Q \leq 1\,000$
2. (10 punten) $M = 2, Q \leq 1\,000$
3. (20 punten) $N, M, Q \leq 100$
4. (26 punten) $Q \leq 5\,000$
5. (35 punten) Geen extra randvoorwaarden.

Sample Grader

De sample grader leest de invoer in volgens het volgende format:

- regel 1: $L \; N \; X \; M \; Q$
- regel 2: $T[0] \; T[1] \; \ldots \; T[N-1]$
- regel 3: $W[0] \; W[1] \; \ldots \; W[N-1]$
- regel 4: $S[0] \; S[1] \; \ldots \; S[M-1]$
- regel $5 + k$ ($0 \leq k < Q$): Y voor vraag k

De sample grader print jouw antwoorden in het volgende format:

- regel $1 + k$ ($0 \leq k < Q$): de return-waarde `arrival_time` voor vraag k