# Overtaking

Budapesht aeroportidan Forras mehmonxonasiga bir qatorli, bir tomonlama yo'l bor. Yo'lning uzunligi $L$ kilometrni tashkil etadi.

IOI 2023 tadbiri davomida $N + 1$ transfer avtobuslari ushbu yo‘ldan o‘tadi. Avtobuslar $0$ dan $N$ gacha raqamlangan. $i$ ($0 \leq i < N$) avtobus aeroportdan tadbirning $T[i]$-inchi soniyasida jo‘nab ketishi rejalashtirilgan va $1$ kilometrni $W[i]$ soniyada bosib o‘tishi mumkin. $N$ - avtobus bu zaxira avtobus bo'lib, $1$ kilometrni $X$ soniyada bosib o'ta oladi. Uning aeroportni qachon tark etish vaqti $Y$ hali aniqlanmagan.

Umuman yo‘lda quvib o‘tish mumkin emas, lekin **saralash stansiyalarida** avtobuslarga bir-biridan o‘tib ketishga ruxsat beriladi. Yo'lda turli pozitsiyalarda $0$ dan $M - 1$ gacha raqamlangan $M$ ($M > 1$) ta saralash stantsiyalari mavjud. $j$ ($0 \leq j < M$) - saralash stantsiyasi aeroportdan $S[j]$ kilometr uzoqlikda yo‘l bo‘ylab joylashgan. Saralash stantsiyalari aeroportdan ortib borayotgan masofada saralanadi, ya'ni har bir $0 \leq j \leq M - 2$ uchun $S[j] < S[j + 1]$. Birinchi saralash stantsiyasi aeroport, oxirgisi esa mehmonxona, ya'ni $S[0] = 0$ va $S[M - 1] = L$.

Har bir avtobus yo'lda o'zidan oldinda sekinroq ketayotgan avtobusga yetib olmasa maksimal tezlikda harakatlanadi, aks holda ular to'planib qoladilar va navbatdagi saralash stantsiyasiga yetib borguncha sekinroq avtobus tezligida harakatlanishga majbur bo'ladilar. U yerda tezroq avtobuslar sekinroq avtobuslardan o'tib ketadi.

Rasmiy ravishda har bir $i$($0 \leq i \leq N$) - avtobus $j$($0 \leq j < M$) - saralash stansiyasiga, **yetib kelish vaqti** $t_{i,j}$ (sekundlarda) quyidagicha aniqlanadi. Har bir $0 \leq i < N$ uchun $t_{i,0} = T[i]$ bo‘lsin hamda $t_{N,0} = Y$ bo‘lsin. Har bir $j$ uchun $0 < j < M$:

- $i$ avtobusning $j$ saralash stansiyasiga **kutilayotgan yetib kelish vaqtini** (soniyalarda) $e_{i,j}$ bilan belgilanadi, bu $i$ avtobusi saralashda yetib borish vaqti sifatida $j$ stansiyasi, agar u $j - 1$ saralash stantsiyasiga kelgan paytdan boshlab to'liq tezlikda harakatlanayotgan bo'lsa:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ har bir $0 \leq i < N$ uchun va
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- $i$ avtobusi $j$ saralash stansiyasiga $i$ avtobusining va $j - 1$ stansiyasiga $i$ dan erta kelgan har bir boshqa avtobusning kutilgan kelishining *maksimalida* keladi. Rasmiy ravishda $t_{i,j}$ maksimal $e_{i,j}$ va har bir har $e_{k,j}$($0 \leq k \leq N$) uchun va $t_{k,j-1} < t_{i,j-1}$ bo'lsin.

IOI tashkilotchilari zahiradagi avtobusni (avtobus $N$) rejalashtirishni xohlashadi. Sizning vazifangiz tashkilotchilarning quyidagi ko'rinishdagi $Q$ ta savollariga javob berishdir: zahiradagi avtobus

aeroportdan $Y$(sekundlarda) vaqtda chiqishi kerak bo'lsa, mehmonxonaga nechanchi vaqtda yetib boradi?

## Implementatsiya tavsilotlari

Sizning vazifangiz quyidagi protsedurani implementatrsiya qilishdan iborat:

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

- $L$: yo'l uzunligi.
- $N$: zaxira bo'lmagan avtobuslar soni.
- $T$: $N$ uzunlikdagi massiv, zaxira bo'lmagan avtobuslarning aeroportdan jo'nab ketishi rejalashtirilgan vaqtlarni tavsiflaydi.
- $W$: zaxira bo'lmagan avtobuslarning maksimal tezligini tavsiflovchi $N$ uzunlikdagi massiv.
- $X$: zahiradagi avtobusning $1$ km masofani bosib o'tish vaqti.
- $M$: saralash stantsiyalari soni.
- $S$: saralash stansiyalarining aeroportdan masofalarini tavsiflovchi $M$ uzunlikdagi massiv.
- Ushbu protsedura har bir test uchun `arrival_time` ga murojaatlardan oldin faqatgina bir marotaba chaqiriladi.

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

- $Y$: zaxira avtobusi (avtobus $N$) aeroportdan ketishi kerak bo'lgan vaqt.
- Ushbu funksiya zaxira avtobusining mehmonxonaga kelishi vaqtini qaytarishi kerak.
- Ushbu funksiyaga aynan $Q$ marotaba murojaat qilinadi.

## Example

Quyidagi murojaatlar ketma-ketligini ko'rib chiqaylik:

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

$4$ avtobusga (hali rejalashtirilmagan) e'tibor bermagan holda, quyidagi jadvalda har bir saralash stantsiyasiga zaxira bo'lmagan avtobuslarning kutilayotgan va haqiqiy kelish vaqtlari ko'rsatilgan:

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

$0$ stansiyasiga yetib kelish vaqti avtobuslarning aeroportni tark etishi rejalashtirilgan vaqtdir. Ya'ni, $0 \le i \le 3$ uchun $t_{i,0} = T[i]$.

$1$ saralash stantsiyasiga kutilayotgan va haqiqiy kelish vaqtlari quyidagicha hisoblanadi:

- The expected times of arrivals at 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 $3$: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- The times of arrivals at station $1$:
  - Buses $1$ and $3$ arrive at station $0$ earlier than bus $0$, so $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Bus $3$ arrives at station $0$ earlier than bus $1$, so $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Bus $0$, bus $1$ and bus $3$ arrive at sorting station $0$ earlier than bus $2$, so $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - No bus arrives at station $0$ earlier than bus $3$, so $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Bus $4$ takes $10$ seconds to travel $1$ kilometre and is now scheduled to leave the airport at the $0$-th second. In this case, the following table shows the times of arrivals for each bus. The only change regarding the expected and actual arrival times of the non-reserve buses is underlined.

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

We see that bus $4$ arrives at the hotel at the $60$-th second. Thus, the procedure should return $60$.

```
arrival_time(50)
```

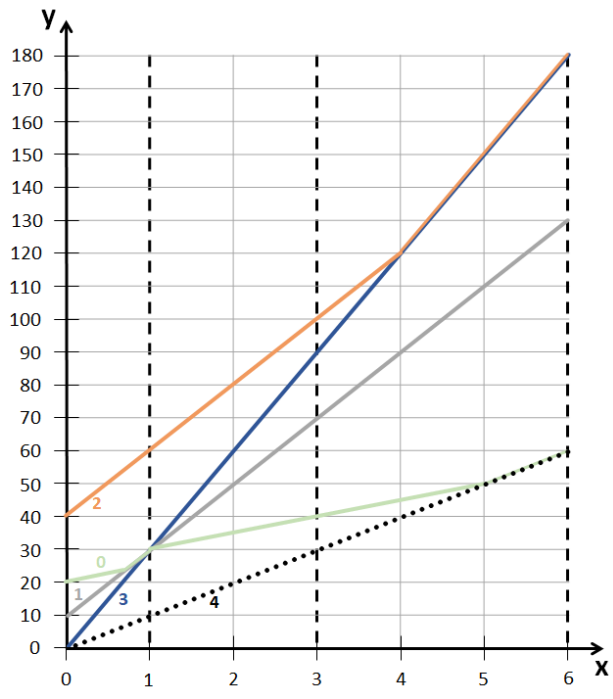
Bus $4$ is now scheduled to leave the airport at the $50$-th second. In this case, there are no changes in the times of arrivals for the non-reserve buses compared to the initial table. The times of arrivals are shown in the following table.

| $i$ | $t_{i,0}$ | $e_{i,1}$ | $t_{i,1}$ | $e_{i,2}$ | $t_{i,2}$ | $e_{i,3}$ | $t_{i,3}$ |
|---|---|---|---|---|---|---|---|
| 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 |

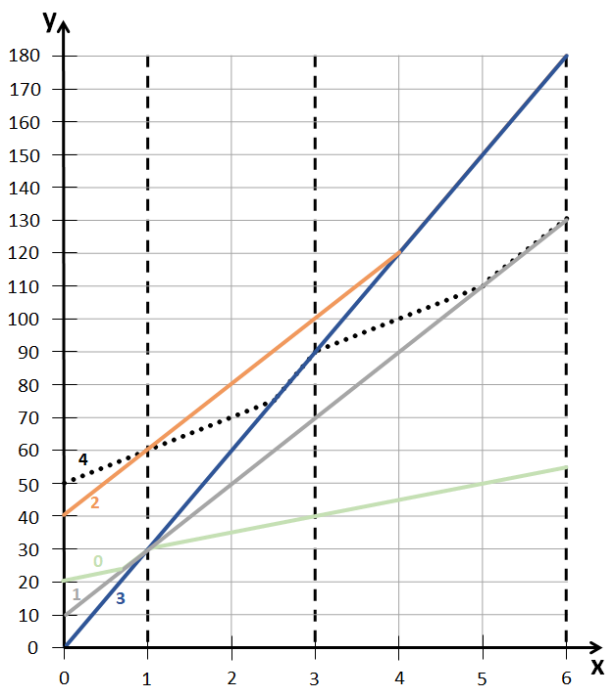
Bus $4$ overtakes the slower bus $2$ at sorting station $1$ as they arrive at the same time. Next, bus $4$ gets bunched with bus $3$ between station $1$ and station $2$, making bus $4$ arrive at station $2$ at the $90$-th second instead of the $80$-th. After leaving station $2$, bus $4$ gets bunched with bus $1$ up until they arrive at the hotel. Bus $4$ arrives at the hotel at the $130$-th second. Thus, the procedure should return $130$.

We can plot the time it takes for each bus to arrive at each distance from the airport. The x-axis of the plot represents the distance from the airport (in kilometres) and the y-axis of the plot represents the time (in seconds). Vertical dashed lines mark the positions of the sorting stations. Different solid lines (accompanied by the bus indices) represent the four non-reserve buses. The dotted black line represents the reserve bus.

**arrival_time(0)**




**arrival_time(50)**




## Constraints

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

## Subtasks

1. (9 points) $N = 1, Q \le 1\,000$
2. (10 points) $M = 2, Q \le 1\,000$
3. (20 points) $N, M, Q \le 100$
4. (26 points) $Q \le 5\,000$
5. (35 points) No additional constraints.

## Sample Grader

The sample grader reads the input in the following format:

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

The sample grader prints your answers in the following format:

- line $1 + k$ ($0 \le k < Q$): the return value of `arrival_time` for question $k$