Overtaking

Между аэропортом Будапешта и отелем Форраш существует односторонняя дорога длиной в L километров.

Во время IOI 2023, $N+1$ автобусов ездят по этой дороге. Автобусы пронумерованы от 0 до N. Автобус номер i ($0 \le i < N$) отправляется из аэропорта в секунду номер $T[i]$, и может ехать со скоростью 1 километр за $W[i]$ секунд. Автобус номер N является резервным и может ехать со скоростью 1 километр за X секунд. Его время отправления Y из аэропорта еще не определено.

По умолчанию обгон на дороге запрещен, но автобусы могут обгонять друг друга на **остановках**. На дороге есть M ($M > 1$) остановок, пронумерованных от 0 до $M-1$, которые расположены в различных позициях. Остановка j ($0 \le j < M$) находится на расстоянии $S[j]$ километров от аэропорта. Остановки упорядочены по расстоянию от аэропорта в порядке возрастания, а именно $S[j] < S[j+1]$ для всех $0 \le j \le M-2$. Первая остановка находится в аэропорту, а последняя в отеле. Таким образом, $S[0] = 0$ и $S[M-1] = L$.

Каждый автобус едет на своей максимальной скорости, кроме тех случаев, когда он догоняет более медленный автобус перед ним. В таком случае, оба автобуса начинают ехать с меньшей из двух скоростей до тех пор, пока не достигнут следующей остановки. На ней более быстрые автобусы обгонят более медленные.

Формально, для всех i и j, таких что $0 \le i \le N$ и $0 \le j < M$, время $t_{i,j}$ (в секундах) когда автобус i **приезжает** на остановку j определено следующим образом. Пусть $t_{i,0} = T[i]$ для всех $0 \le i < N$, и пусть $t_{N,0} = Y$. Для всех j, таких что $0 < j < M$:

- Определим **ожидаемое время прибытия** (в секундах) автобуса i на остановку j, обозначенное за $e_{i,j}$, как время когда автобус i приедет на остановку j если бы он ехал на полной скорости начиная с момента, как он приехал на остановку $j-1$. А именно, пусть
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ для всех $0 \le i < N$, и
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Автобус i приедет на остановку j в момент, равный *максимуму* из ожидаемых времен прибытия автобуса i и всех остальных автобусов, которые приехали на остановку $j-1$ раньше автобуса i. Формально, значение $t_{i,j}$ равно максимуму из $e_{i,j}$ и всех $e_{k,j}$, для которых $0 \le k \le N$ и $t_{k,j-1} < t_{i,j-1}$.

Организаторы IOI хотят запланировать отправление резервного автобуса (под номером N). Вашей задачей является ответить на Q запросов от организаторов, имеющих следующую форму: дано время Y (в секундах) когда резервный автобус должен покинуть аэропорт, в какое время он приедет в отель?

Implementation Details

Вы должны реализовать следующие функции.

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

- L: длина дороги.
- N: количество нерезервных автобусов.
- T: массив длины N, задающий время отправления нерезевных автобусов из аэропорта.
- W: массив длины N, задающий максимальную скорость нерезевных автобусов.
- X: время, необходимое резервному автобусу для проезда 1 километра.
- M: количество остановок.
- S: массив длины M, задающий расстояния от аэропорта до остановок.
- Эта функция будет вызвана ровно один раз в каждом тесте, перед всеми вызовами функции `arrival_time`.

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

- Y: время, когда резервный автобус должен выехать из аэропорта.
- Эта функция должна вернуть время, в которое резервный автобус приедет в отель.
- Эта функция будет вызвана ровно Q раз.

Example

Рассмотрим следующую серию вызовов:

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

Игнорируя автобус 4 (который еще не был запланирован), следующая таблица показывает ожидаемые и реальные времена прибытия нерезервных автобусов на каждую остановку:

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

Время прибытия на остановку 0 в таблице равно запланированному времени отправления автобуса. То есть, $t_{i,0} = T[i]$ для $0 \le i \le 3$.

Ожидаемые и реальные времена прибытия на остановку 1 вычисляются следующим образом:

- Ожидаемое время прибытия на остановку 1:
  - Автобус 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Автобус 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Автобус 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Автобус 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Реальное время прибытия на остановку 1:
  - Автобусы 1 и 3 приедут на остановку 0 раньше автобуса 0, поэтому $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Автобус 3 приедет на остановку 0 раньше автобуса 1, поэтому $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Автобусы 0, 1 и 3 приедут на остановку 0 раньше автобуса 2, поэтому $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Ни один автобус не приедет на остановку 0 раньше автобуса 3, поэтому $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Автобус 4 проезжает 1 километр за 10 секунд, он должен покинуть аэропорт в секунду 0. В этом случае, следующая таблица показывает времена прибытия каждого автобуса. Все изменения времен прибытия нерезервных автобусов подчеркнуты.

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

Как видно выше, автобус 4 приезжает в отель в 60-ю секунду. Поэтому функция должна вернуть 60.

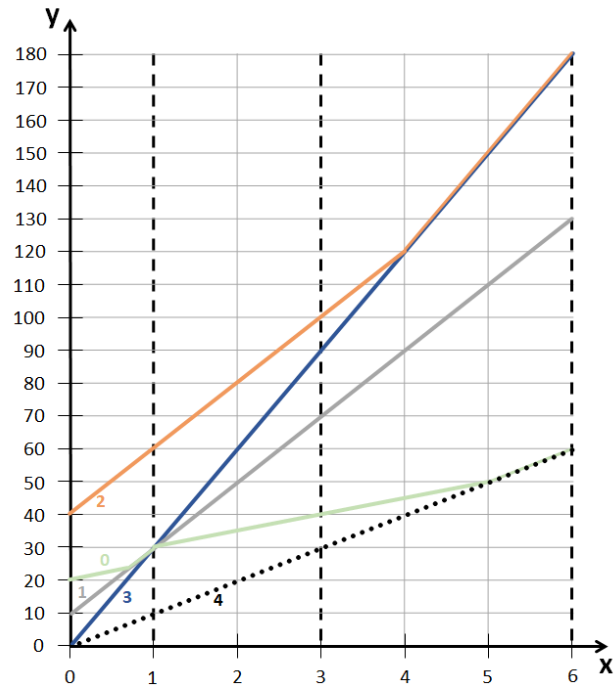
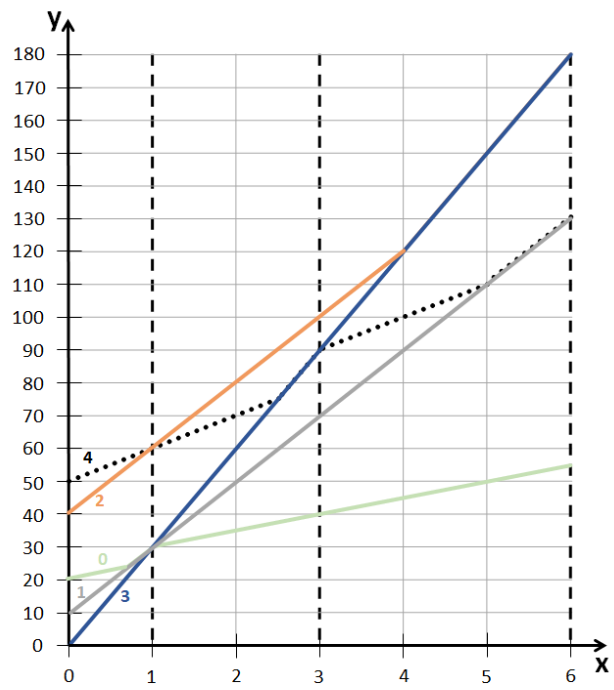
```
arrival_time(50)
```

Теперь автобус 4 должен покинуть аэропорт в секунду 50. В этом случае, в сравнении с изначальной таблицей, времени прибытия не поменяются для нерезервных автобусов. Времена прибытия приведены в таблице ниже.

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

Автобус 4 обгоняет более медленный автобус 2 на остановке 1, поскольку они приезжают в одно время. Далее, автобус 4 догоняет автобус 3 между остановками 1 и 2, из-за чего автобус 4 приезжает на остановку 2 в 90-ю секунду вместо 80-й. После отправления с остановки 2, автобус 4 догоняет автобус 1, после чего они вместе доезжают до отеля. Автобус 4 приезжает в отель в секунду 130. Поэтому функция должна вернуть значение 130.

Мы можем нарисовать график, сколько времени требуется каждому автобусу чтобы проехать определенное расстояние от аэропорта. Ось Ox графика обозначает расстояние от аэропорта (в километрах), а ось Oy обозначает время (в секундах). Вертикальные пунктирные линии обозначают позиции остановок. Различные сплошные линии (помеченные индексами автобусов) обозначают четыре нерезервных автобуса. Линия из точек обозначает резервный автобус.

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

Constraints

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

Subtasks

1. (9 баллов) $N = 1, Q \le 1\,000$
2. (10 баллов) $M = 2, Q \le 1\,000$
3. (20 баллов) $N, M, Q \le 100$
4. (26 баллов) $Q \le 5\,000$
5. (35 баллов) Нет дополнительных ограничений.

Sample Grader

Грейдер читает входные данные в следующем формате:

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

Грейдер выводит ваши ответы в следующем формате:

- строка $1+k$ ($0 \le k < Q$): возвращаемое значение `arrival_time` для запроса k