



Overtaking

There is a one-lane, one-way road from Budapest Airport to Hotel Forrás. The road is L kilometres long.

Over the IOI 2023 event, $N + 1$ transfer buses traverse this road. Buses are numbered from 0 to N . Bus i ($0 \leq i < N$) is scheduled to leave the airport at the $T[i]$ -th second of the event, and can travel 1 kilometre in $W[i]$ seconds. Bus N is a reserve bus that can travel 1 kilometre in X seconds. The time Y when it will leave the airport has not yet been decided.

Overtaking is not allowed on the road in general, but the buses are allowed to overtake each other at **sorting stations**. There are M ($M > 1$) sorting stations, numbered from 0 to $M - 1$, on different positions on the road. Sorting station j ($0 \leq j < M$) is located $S[j]$ kilometres from the airport along the road. The sorting stations are sorted in increasing distance from the airport, that is, $S[j] < S[j + 1]$ for each $0 \leq j \leq M - 2$. The first sorting station is the airport and the last one is the hotel, that is, $S[0] = 0$ and $S[M - 1] = L$.

Each bus travels at maximum speed unless it catches up to a slower bus travelling ahead of it on the road, in which case they get bunched and forced to travel at the speed of the slower bus, until they reach the next sorting station. There, the faster buses will overtake the slower buses.

Formally, for each i and j such that $0 \leq i \leq N$ and $0 \leq j < M$, the time $t_{i,j}$ (in seconds) when bus i **arrives at** sorting station j is defined as follows. Let $t_{i,0} = T[i]$ for each $0 \leq i < N$, and let $t_{N,0} = Y$. For each j such that $0 < j < M$:

- Define the **expected time of arrival** (in seconds) of bus i at sorting station j , denoted by $e_{i,j}$, as the time when bus i would arrive at sorting station j if it was travelling at full speed from the time it arrived at sorting station $j - 1$. That is, let
 - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ for each $0 \leq i < N$, and
 - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$.
- Bus i arrives at sorting station j at the *maximum* of the expected times of arrivals of bus i and of every other bus that arrived at station $j - 1$ earlier than bus i . Formally, let $t_{i,j}$ be the maximum of $e_{i,j}$ and every $e_{k,j}$ for which $0 \leq k \leq N$ and $t_{k,j-1} < t_{i,j-1}$.

The IOI organizers want to schedule the reserve bus (bus N). Your task is to answer Q questions of the organizers, which are of the following form: given the time Y (in seconds) when the reserve bus is supposed to leave the airport, at what time would it arrive at the hotel?

Implementation Details

Your task is to implement the following procedures.

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

- L : the length of the road.
- N : the number of non-reserve buses.
- T : an array of length N describing the times at which non-reserve buses are scheduled to leave from the airport.
- W : an array of length N describing the maximum speeds of non-reserve buses.
- X : the time it takes for the reserve bus to travel 1 kilometre.
- M : the number of sorting stations.
- S : an array of length M describing the distances of the sorting stations from the airport.
- This procedure is called exactly once for each test case, before any calls to `arrival_time`.

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

- Y : the time at which the reserve bus (bus N) is supposed to leave from the airport.
- This procedure should return the time at which the reserve bus would arrive at the hotel.
- This procedure is called exactly Q times.

Example

Consider the following sequence of calls:

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

Ignoring bus 4 (that has not yet been scheduled), the following table shows the expected and actual times of arrivals for non-reserve buses at each sorting station:

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

The times of arrivals at station 0 are the times at which buses are scheduled to leave the airport. That is, $t_{i,0} = T[i]$ for $0 \leq i \leq 3$.

The expected and actual times of arrivals at sorting station 1 are computed as follows:

- 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	<u>60</u>
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.

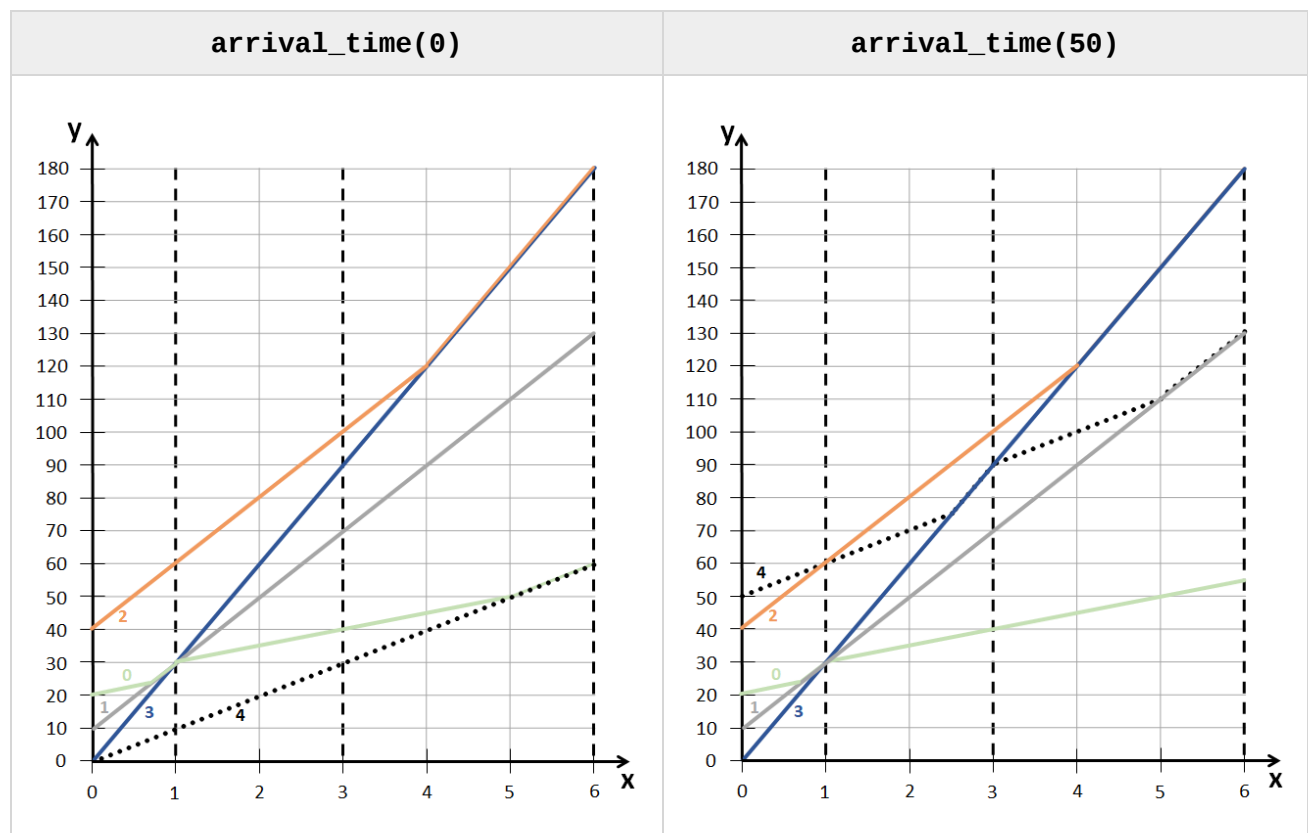
```
arrival_time(50)
```

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.



Constraints

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

Subtasks

1. (9 points) $N = 1, Q \leq 1\,000$
2. (10 points) $M = 2, Q \leq 1\,000$
3. (20 points) $N, M, Q \leq 100$
4. (26 points) $Q \leq 5\,000$
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: $T[0]\ T[1]\ \dots\ T[N - 1]$
- line 3: $W[0]\ W[1]\ \dots\ W[N - 1]$
- line 4: $S[0]\ S[1]\ \dots\ S[M - 1]$
- line $5 + k$ ($0 \leq k < Q$): Y for question k

The sample grader prints your answers in the following format:

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