



추월

부다페스트 공항에서 포라스 호텔까지는 차선이 하나만 있는 일방통행 도로가 있다. 이 도로의 길이는 L 킬로미터이다.

IOI 2023 행사 기간 동안, $N + 1$ 대의 버스가 이 도로를 지난다. 버스는 0부터 N 까지 번호가 붙는다. 버스 i ($0 \leq i < N$)는 행사 시작 $T[i]$ 초 뒤에 공항을 떠날 계획이고, $W[i]$ 초 동안 1 킬로미터를 갈 수 있다. 버스 N 은 X 초 동안 1 킬로미터를 갈 수 있는 예비 버스이다. 이 버스가 공항을 떠나는 시간 Y 는 아직 정해지지 않았다.

기본적으로 도로에서 추월은 허락되지 않지만, **추월 장소**에서는 추월이 가능하다. 0부터 $M - 1$ 까지 번호가 붙은 M ($M > 1$)개의 추월 장소가 도로의 서로 다른 위치에 있다. 추월 장소 j ($0 \leq j < M$)는 공항에서 도로를 따라 $S[j]$ 킬로미터 거리에 있다. 추월 장소들은 공항으로부터 거리의 증가순으로 정렬되어 있다. 즉, 각 $0 \leq j \leq M - 2$ 에 대해 $S[j] < S[j + 1]$ 이다. 첫번째 추월 장소는 공항이고 마지막 추월 장소는 호텔이다. 즉, $S[0] = 0$ 이고 $S[M - 1] = L$ 이다.

각 버스는 앞서 가는 느린 버스를 따라잡지 못했으면 최대 속도로 움직이지만, 따라잡으면 다음 추월 장소에 도달할 때까지는 느린 버스의 속도로 함께 움직인다. 추월 장소에서, 더 빠른 버스들이 더 느린 버스들을 추월할 것이다.

엄밀하게, $0 \leq i \leq N$ 이고 $0 \leq j < M$ 인 각 i 와 j 에 대해, 버스 i 가 추월 장소 j 에 **도착하는** 시간 $t_{i,j}$ (초)는 다음과 같이 정의된다. 각 $0 \leq i < N$ 에 대해 $t_{i,0} = T[i]$ 로 두고 $t_{N,0} = Y$ 로 둔다. $0 < j < M$ 인 각 j 에 대해:

- 버스 i 의 추월 장소 j 에의 **도착 예정 시간**(초) $e_{i,j}$ 는 버스 i 가 추월 장소 $j - 1$ 에 도착한 시간으로부터 최대 속도로 이동한 경우에 추월 장소 j 에 도착하는 시간으로 정의한다. 즉,
 - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ 로 두고(각 $0 \leq i < N$ 에 대해),
 - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$ 로 둔다.
- 버스 i 가 추월 장소 j 에 도착하는 시간은, 추월 장소 $j - 1$ 에서 버스 i 보다 앞서 도착했던 버스들, 그리고 버스 i 의 도착 예정 시간의 **최대** 시간에 도착한다. 엄밀하게, $t_{i,j}$ 를 $e_{i,j}$ 와 $0 \leq k \leq N$ 이고 $t_{k,j-1} < t_{i,j-1}$ 인 모든 $e_{k,j}$ 의 최댓값으로 둔다.

IOI 운영진은 예비 버스(버스 N)의 출발 시간을 결정하고 싶다. 당신의 작업은 운영진이 하는 Q 개의 질문에 답하는 것이다. 질문은 다음과 같은 형식이다: 예비 버스가 공항을 떠나는 시간을 Y (초)로 가정할 때, 예비 버스가 호텔에 도착하는 시간이 언제인가?

Implementation Details

다음 함수를 구현해야 한다.

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

- L : 도로의 길이.
- N : 예비가 아닌 버스 수.
- T : 예비가 아닌 버스들이 공항을 떠나도록 계획된 시간을 나타내는 길이 N 인 배열.
- W : 예비가 아닌 버스들의 최대 속도를 나타내는 길이 N 인 배열.
- X : 예비 버스가 1 킬로미터를 가는데 걸리는 시간.
- M : 추월 장소의 수.
- S : 공항에서 각 추월 장소까지의 거리를 나타내는 길이 M 인 배열.
- 이 함수는 `arrival_time`이 호출되기 전에 각 테스트 케이스에 대해 정확히 한번만 호출된다.

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

- Y : 예비 버스(버스 N)가 공항을 떠나는 것으로 가정하는 시간.
- 이 함수는 예비 버스가 호텔에 도착하는 시간을 리턴해야 한다.
- 이 함수는 정확히 Q 번 호출된다.

Example

다음 호출들을 생각해보자:

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

아직 출발 시간이 정해지지 않은 버스 4를 무시하면, 다음 표는 예비가 아닌 버스들의 각 추월 장소에서의 도착 예정 시간과 실제 도착 시간을 보여준다:

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

장소 0에 도착하는 시간은 버스가 공항을 떠나기로 계획된 시간이다. 즉, $0 \leq i \leq 3$ 에 대해 $t_{i,0} = T[i]$ 이다.

추월 장소 1의 도착 예정 시간과 실제 도착 시간은 다음과 같이 계산된다:

- 장소 1의 도착 예정 시간:
 - 버스 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
 - 버스 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
 - 버스 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
 - 버스 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- 장소 1의 도착 시간:
 - 버스 1과 3은 장소 0에 버스 0보다 먼저 도착하므로, $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
 - 버스 3은 장소 0에 버스 1보다 먼저 도착하므로, $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
 - 버스 0, 버스 1, 버스 3은 추월 장소 0에 버스 2보다 먼저 도착하므로, $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
 - 장소 0에 버스 3보다 먼저 도착하는 버스는 없으므로, $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

버스 4는 1킬로미터를 움직이는데 10초가 걸리고, 0초에 공항을 떠나는 것으로 가정된다. 이 경우, 다음 표는 각 버스의 도착 시간을 보여준다. 예비가 아닌 버스들의 도착 예정 시간과 실제 도착 시간의 유일한 변화는 밑줄이 쳐져 있다.

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

버스 4는 호텔에 60초에 도착함을 알 수 있다. 따라서 이 함수는 60을 리턴해야 한다.

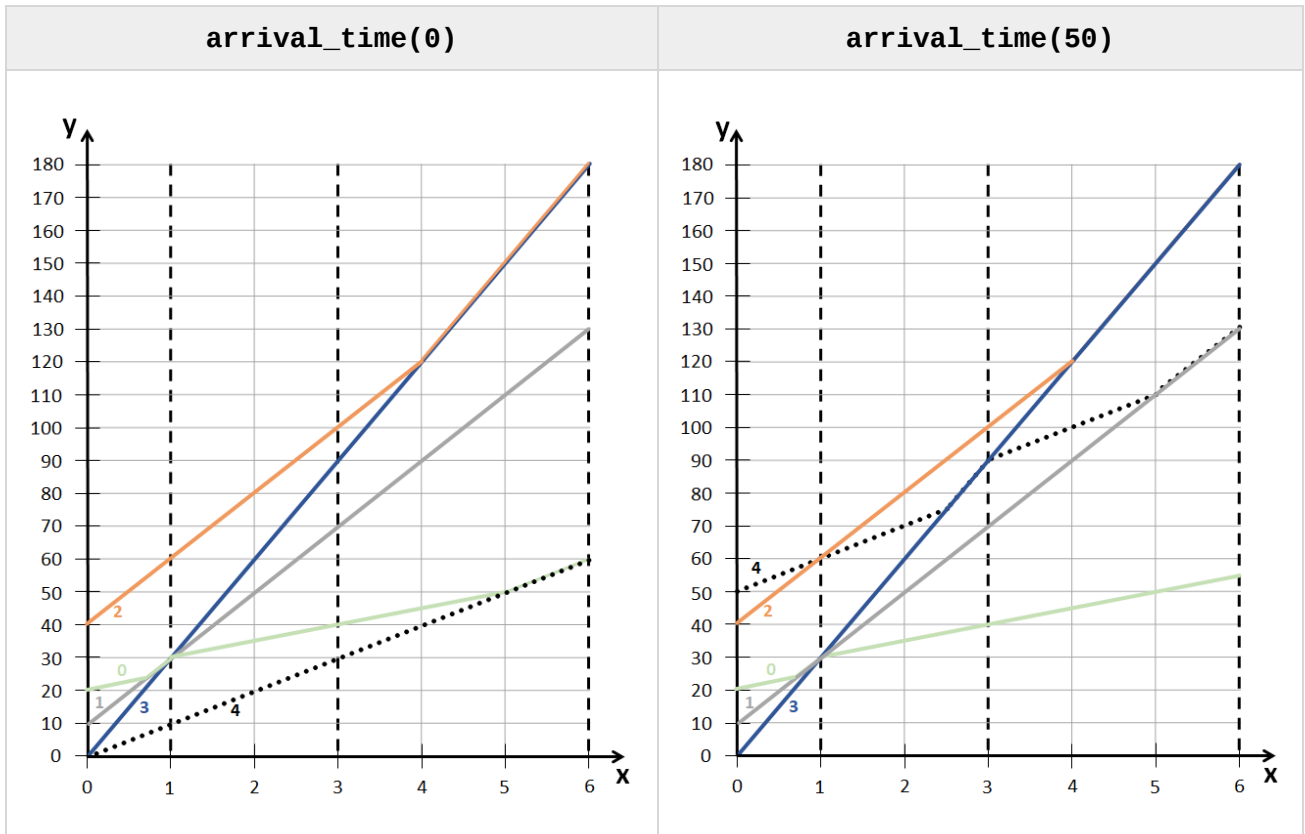
```
arrival_time(50)
```

버스 4는 50초에 공항을 떠나는 것으로 가정된다. 이 경우, 예비가 아닌 버스들의 도착 시간은 첫 표와 비교해서 변화가 없다. 도착 시간은 다음 표와 같다.

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

버스 4는 느린 버스 2를 돌이 동시에 도착하는 추월 장소 1에서 추월한다. 다음, 버스 4는 장소 1과 장소 2 사이에서 버스 3과 함께 움직이게 되어, 버스 4가 장소 2에 도착하는 시간이 80초에서 90초로 바뀐다. 장소 2를 떠난 후에, 버스 4는 호텔에 도착할 때까지 버스 1과 함께 움직이게 된다. 버스 4는 호텔에 130초에 도착한다. 따라서, 이 함수는 130을 리턴해야 한다.

공항에서의 거리에 따라 각 버스가 도착하는 시간을 그래프로 그릴 수 있다. x축은 공항으로부터의 거리(킬로미터)를 나타내고 y축은 시간(초)을 나타낸다. 세로 점선은 추월 장소를 표시한다. (버스 번호가 같이 적힌) 서로 다른 실선은 네 대의 예비가 아닌 버스를 나타낸다. 검정 점선은 예비 버스를 나타낸다.



Constraints

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1000$
- $0 \leq T[i] \leq 10^{18}$ ($0 \leq i < N$ 인 각 i 에 대해)
- $1 \leq W[i] \leq 10^9$ ($0 \leq i < N$ 인 각 i 에 대해)
- $1 \leq X \leq 10^9$
- $2 \leq M \leq 1000$
- $0 = S[0] < S[1] < \dots < S[M - 1] = L$
- $1 \leq Q \leq 10^6$
- $0 \leq Y \leq 10^{18}$

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) 추가적인 제한이 없다.

Sample Grader

샘플 그레이더의 입력 형식은 다음과 같다.

- 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 (질문 k 에 대한)

샘플 그레이더는 다음 형식으로 답을 출력한다:

- line $1 + k$ ($0 \leq k < Q$): 질문 k 에 대한 `arrival_time`의 리턴값