Overtaking

Có một con đường một làn, một chiều từ sân bay Budapest đến khách sạn Forrás. Con đường dài L km.

Trong sự kiện IOI 2023, $N+1$ xe buýt đi qua con đường này. Các xe buýt được đánh số từ 0 đến N. Theo kế hoạch xe buýt i ($0 \le i < N$) rời sân bay vào giây thứ $T[i]$ của sự kiện và có thể di chuyển 1 km trong $W[i]$ giây. Xe buýt N là xe buýt dự phòng có thể di chuyển 1 km trong X giây. Thời điểm Y là thời điểm xe buýt dự phòng rời sân bay và nó chưa được xác định.

Nói chung không được phép vượt trên đường, nhưng các xe buýt được phép vượt nhau tại **trạm phân loại**. Có M trạm phân loại ($M > 1$), được đánh số từ 0 đến $M-1$, nằm ở các vị trí khác nhau trên đường. Trạm phân loại j ($0 \le j < M$) ở vị trí $S[j]$ km tính từ sân bay dọc theo con đường. Các trạm phân loại được sắp xếp theo khoảng cách tăng dần tính từ sân bay, nghĩa là $S[j] < S[j+1]$ cho mỗi $0 \le j \le M-2$. Trạm phân loại đầu tiên là sân bay và trạm cuối cùng là khách sạn, nghĩa là $S[0] = 0$ và $S[M-1] = L$.

Mỗi chiếc xe buýt di chuyển với tốc độ tối đa trừ khi nó đuổi kịp một chiếc xe buýt chậm hơn đang di chuyển phía trước trên đường, trong trường hợp đó, chúng sẽ bị dồn lại và buộc phải di chuyển với tốc độ của chiếc xe buýt chậm hơn cho đến khi chúng đến trạm phân loại tiếp theo. Ở đó, xe buýt nhanh hơn sẽ vượt qua xe buýt chậm hơn.

Cụ thể, với mỗi i và j sao cho $0 \le i \le N$ và $0 \le j < M$, thời điểm $t_{i,j}$ (tính bằng giây) khi xe buýt i **đến** trạm phân loại j được xác định như sau. Đặt $t_{i,0} = T[i]$ cho mỗi $0 \le i < N$, và đặt $t_{N,0} = Y$. Với mỗi j mà $0 < j < M$:

- Xác định **thời điểm dự kiến đến** (tính bằng giây) của xe buýt i tại trạm phân loại j, kí hiệu là $e_{i,j}$, là thời điểm xe buýt i sẽ đến trạm phân loại j nếu nó chạy với tốc độ tối đa kể từ thời điểm đến trạm phân loại $j-1$. Tức là, đặt
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ cho mỗi $0 \le i < N$, và
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- Xe buýt i đến trạm phân loại j tại thời điểm *lớn nhất* trong các thời điểm dự kiến đến của xe buýt i và của mọi xe buýt khác đã đến trạm $j-1$ sớm hơn xe buýt i. Cụ thể, đặt $t_{i,j}$ là giá trị lớn nhất của $e_{i,j}$ và mọi $e_{k,j}$ mà $0 \le k \le N$ và $t_{k,j-1} < t_{i,j-1}$.

Ban tổ chức IOI muốn lên lịch cho xe buýt dự phòng (xe buýt N). Nhiệm vụ của bạn là trả lời Q câu hỏi của ban tổ chức, có dạng sau: Cho thời điểm Y (tính bằng giây) khi xe buýt dự phòng rời sân bay thì xe sẽ đến khách sạn vào thời điểm nào?

Chi tiết cài đặt

Bạn cần cài đặt các hàm sau:

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

- L: chiều dài của con đường.
- N: số lượng xe buýt không phải xe dự phòng.
- T: một mảng kích thước N mô tả thời điểm mà các xe buýt (không phải xe dự phòng) rời khỏi sân bay theo kế hoạch.
- W: mảng kích thước N mô tả tốc độ tối đa của các xe buýt (không phải xe dự phòng).
- X: thời gian để xe buýt dự phòng đi được 1 km.
- M: số trạm phân loại.
- S: mảng kích thước M mô tả khoảng cách vị trí các trạm phân loại tính từ sân bay.
- Hàm này được gọi đúng một lần cho mỗi test, trước bất kì lời gọi nào đến `arrival_time`.

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

- Y: thời điểm xe buýt dự phòng (xe buýt N) khởi hành từ sân bay.
- Hàm này sẽ trả về thời điểm xe buýt dự phòng đến khách sạn.
- Hàm này được gọi đúng Q lần.

Ví dụ

Xét dãy các lời gọi sau:

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

Bỏ qua xe buýt 4 (chưa được lên lịch), bảng sau đây cho biết thời điểm đến dự kiến và thực tế của các xe buýt không phải xe dự phòng tại mỗi trạm phân loại:

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

Thời điểm đến trạm 0 là thời điểm xe buýt rời sân bay theo kế hoạch. Nghĩa là, $t_{i,0} = T[i]$ với $0 \le i \le 3$.

Thời điểm dự kiến và thực tế của các xe đến trạm phân loại 1 được tính như sau:

- Thời điểm dự kiến đến trạm 1:
  - Xe buýt 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Xe buýt 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Xe buýt 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Xe buýt 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Thời điểm đến trạm 1:
  - Xe buýt 1 và 3 đến trạm 0 sớm hơn xe buýt 0, nên $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - Xe buýt 3 đến trạm 0 sớm hơn xe buýt 1, nên $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Xe buýt 0, xe buýt 1 và xe buýt 3 đến trạm phân loại 0 sớm hơn xe buýt 2, vì vậy $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Không có xe buýt nào đến trạm 0 sớm hơn xe buýt 3, nên $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Xe buýt 4 mất 10 giây để đi hết 1 km và rời sân bay vào giây thứ 0. Trong trường hợp này, bảng sau đây hiển thị thời điểm đến của mỗi xe buýt. Thay đổi duy nhất liên quan đến thời điểm đến dự kiến và thực tế của xe buýt (không phải xe dự phòng) được gạch chân.

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

Chúng ta thấy rằng xe buýt 4 đến khách sạn ở giây thứ 60. Vì vậy, hàm sẽ trả về 60.

```
arrival_time(50)
```

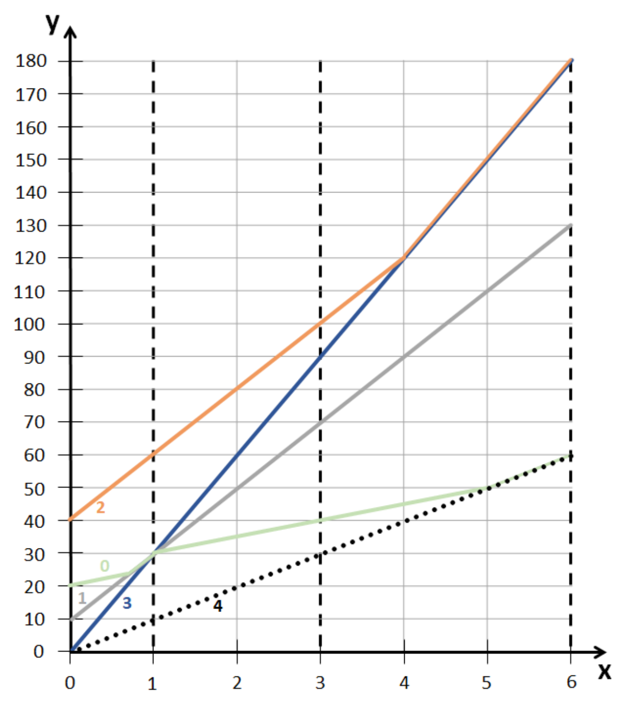
Xe buýt 4 được lên kế hoạch rời sân bay vào giây thứ 50. Trong trường hợp này, không có sự thay đổi về thời điểm đến của các xe buýt (không phải xe dự phòng) so với bảng ban đầu. Thời điểm đến được thể hiện trong bảng sau.

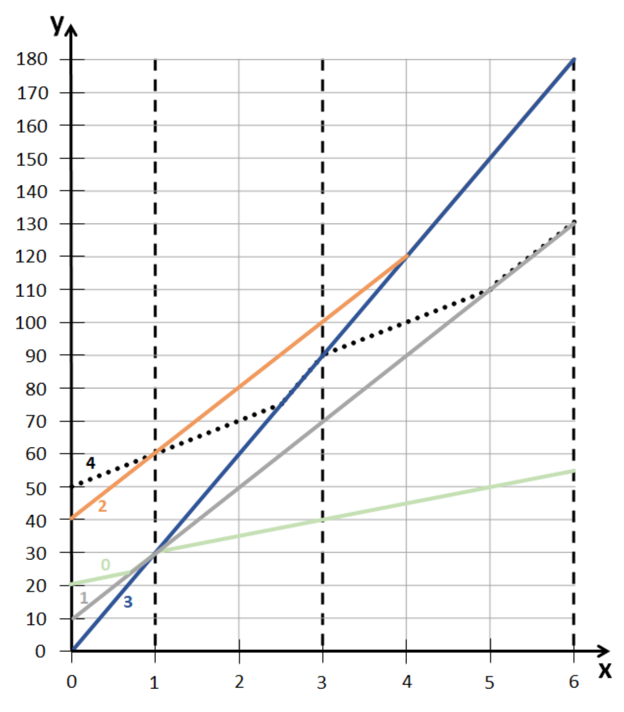
i		$t_{i,0}$		$e_{i,1}$	$t_{i,1}$		$e_{i,2}$	$t_{i,2}$		$e_{i,3}$	$t_{i,3}$
0		20		25	30		40	40		55	55
1		10		30	30		70	70		130	130
2		40		60	60		100	100		160	180
3		0		30	30		90	90		180	180
4		50		60	60		80	90		120	130

Xe buýt 4 vượt qua xe buýt 2 chậm hơn tại trạm phân loại 1 khi chúng đến cùng lúc. Tiếp theo, xe buýt 4 được dồn lại với xe buýt 3 giữa trạm 1 và trạm 2, khiến xe buýt 4 đến trạm 2 ở giây thứ 90 thay vì giây thứ 80. Sau khi rời trạm 2, xe buýt 4 dồn lại với xe buýt 1 cho đến khi chúng đến khách sạn. Xe buýt 4 đến khách sạn ở giây thứ 130. Vì vậy, hàm sẽ trả về 130.

Chúng ta có thể vẽ đồ thị thời gian của mỗi xe buýt phải đi cho mỗi khoảng cách tính từ sân bay. Trục x của đồ thị biểu thị khoảng cách từ sân bay (tính bằng km) và trục y của đồ thị biểu thị thời điểm (tính bằng giây). Các đường nét đứt dọc đánh dấu vị trí các trạm phân loại. Các đường liền nét khác nhau (kèm theo các chỉ số xe buýt) thể hiện bốn xe buýt (không phải xe dự phòng). Đường màu đen chấm thể hiện cho xe buýt dự phòng.

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





Các ràng buộc

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

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

Các subtask

1. (9 điểm) $N = 1, Q \le 1\,000$
2. (10 điểm) $M = 2, Q \le 1\,000$
3. (20 điểm) $N, M, Q \le 100$
4. (26 điểm) $Q \le 5\,000$
5. (35 điểm) Không có ràng buộc nào thêm.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu vào theo định dạng sau:

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

Trình chấm mẫu ghi câu trả lời của bạn theo định dạng sau:

- dòng $1 + k$ ($0 \le k < Q$): giá trị trả về của `arrival_time` cho câu hỏi k