



超車 (Overtaking)

從布達佩斯到 Forrás 飯店有一條道路，此道路為單行道並只有一線車道。這條路的長度為 L 公里。

在 IOI 2023 大會期間，有 $N + 1$ 台巴士行駛於此道路上。巴士編號由 0 至 N 。巴士 i ($0 \leq i < N$) 被安排於大會的第 $T[i]$ 秒離開機場，每行駛 1 公里用 $W[i]$ 秒。巴士 N 為預備巴士，每行駛 1 公里用 X 秒。它離開機場的時刻 Y 尚未被決定。

一般來說，在此道路上超車是不被允許的，但在 **分類站** (sorting stations) 巴士是被允許互相超車的。有 M 個分類站 ($M > 1$) 位於此道路上的不同位置，編號由 0 至 $M - 1$ 。沿著此道路，分類站 j ($0 \leq j < M$) 位於離機場 $S[j]$ 公里處。分類站以離機場的距離遞增排序，即 $S[j] < S[j + 1]$ ，對所有 $0 \leq j \leq M - 2$ 。第一個分類站為機場，最後一個分類站為飯店，即 $S[0] = 0$ 以及 $S[M - 1] = L$ 。

每台巴士皆以其最大速度行駛，除非它追趕上一台駛於其前方且速度較慢的巴士。在此情形下，它們會行駛在一起並被強制以其中較慢的速度前進，直到到達下一個分類站。在那裡較快的巴士將超越較慢的巴士。

正式地說，對每對 i 與 j 滿足 $0 \leq i \leq N$ 且 $0 \leq j < M$ ，巴士 i 到達分類站 j 的時刻 $t_{i,j}$ (單位為秒) 定義如下。對任一 $0 \leq i < N$ ， $t_{i,0} = T[i]$ ，且 $t_{N,0} = Y$ 。對任一 j 滿足 $0 < j < M$ ：

- 定義巴士 i 到達分類站 j 的 **預期到達時刻** (單位為秒)，以 $e_{i,j}$ 表示，為巴士 i 到達分類站 $j - 1$ 的時刻起算，以最大速度行駛至分類站 j 的時刻。意即，
 - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j - 1])$ 對任意 $0 \leq i < N$ ，且
 - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j - 1])$ 。
- 巴士 i 到達分類站 j 的時刻，為巴士 i 以及比巴士 i 早到達分類站 $j - 1$ 的所有巴士中，預期到達時刻的 **最大值**。正式地說， $t_{i,j}$ 為 $e_{i,j}$ 與所有滿足 $0 \leq k \leq N$ 且 $t_{k,j-1} < t_{i,j-1}$ 的 k 之 $e_{k,j}$ 中的最大值。

IOI 的主辦單位想要安排預備巴士 (巴士 N)。你的任務是回答 Q 個主辦單位的問題，問題形式如下：若預備巴士於時刻 Y (單位為秒) 離開機場，該巴士到達飯店的時刻為何？

實作細節 (Implementation Details)

你的任務是實作下列所有程序：

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

- L ：此道路的長度。
- N ：非預備巴士的數量。

- T ：一長度為 N 的陣列，用來描述非預備巴士被安排離開機場的時刻。
- W ：一長度為 N 的陣列，用來描述非預備巴士的最大速度。
- X ：預備巴士行駛 1 公里所需的時間。
- M ：分類站的數量。
- S ：一長度為 M 的陣列，用來描述分類站距離機場的距離。
- 對於每筆測資，在任何 arrival_time 的呼叫前，此程序被呼叫恰好一次。

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

- Y ：預備巴士 (巴士 N) 應離開機場的時刻。
- 此程序應回傳預備巴士到達飯店的時刻。
- 此程序會被呼叫恰好 Q 次。

範例 (Example)

考慮下列呼叫序列：

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

在忽略巴士 4 (因其尚未被安排) 的狀況下，下表展示了非預備巴士於各個分類站的預期到達時刻以及實際到達時刻：

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

各巴士到達分類站 0 的時刻為各巴士被安排離開機場的時刻。即 $t_{i,0} = T[i]$ ，其中 $0 \leq i \leq 3$ 。

各巴士於分類站 1 的預期到達時刻與實際到達時刻的計算方式如下：

- 各巴士於分類站 1 的預期到達時刻：
 - 巴士 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$ 。
 - 巴士 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$ 。
 - 巴士 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$ 。
 - 巴士 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$ 。
- 各巴士於分類站 1 的實際到達時刻：
 - 巴士 1 與 3 較巴士 0 早到達分類站 0，故 $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$ 。
 - 巴士 3 較巴士 1 早到達分類站 0，故 $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$ 。
 - 巴士 0、巴士 1 及巴士 3 較巴士 2 早到達分類站 0，故 $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$ 。

- 沒有巴士較巴士 3 早到達分類站 0，故 $t_{3,1} = \max([e_{3,1}]) = 30$ 。

```
arrival_time(0)
```

巴士 4 耗費 10 秒行駛 1 公里且目前被安排於第 0 秒離開機場。在此情形，下表展示了各巴士的實際到達時刻。非預備巴士的預期到達時間與實際到達時間有差異者以底線標註。

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

可看到巴士 4 於第 60 秒到達飯店。因此，此程序應回傳 60。

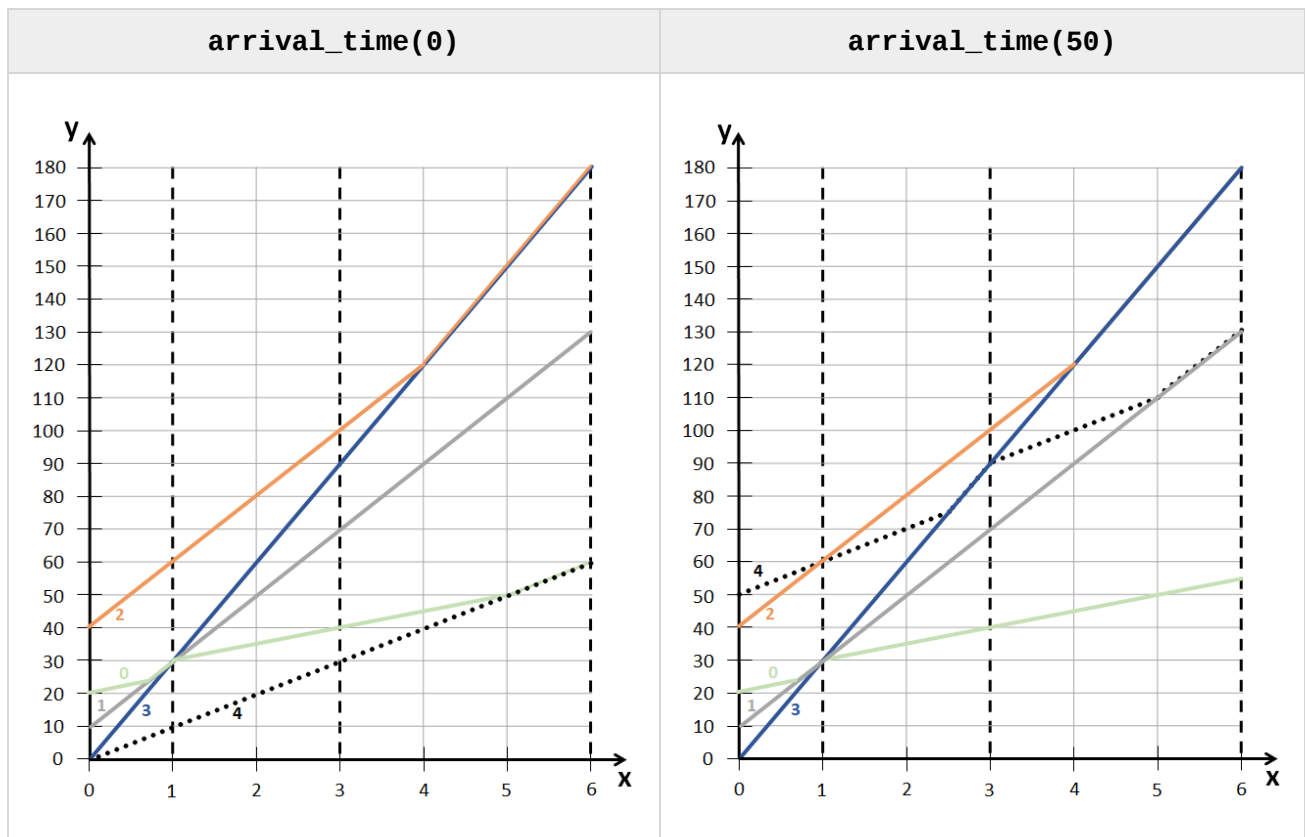
```
arrival_time(50)
```

巴士 4 目前被安排於第 50 秒離開機場。在此情形下，與初始的表格相比非預備巴士的到達時間沒有任何改變。到達時間展示於下表。

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

巴士 4 於分類站 1 超越巴士 2 因為它們同時到達該站。接下來，巴士 4 於分類站 1 與 2 之間和巴士 3 會合，導致巴士 4 於第 90 秒到達分類站 2 而不是於第 80 秒到達。在離開分類站 2 後，巴士 4 與巴士 1 會合直到到達飯店。巴士 4 於第 130 秒到達飯店。因此，此程序應回傳 130。

我們可繪製每台巴士由機場開始到達各距離的時間。x-軸表示與機場的距離(單位為公里)而 y-軸表示時間(單位為秒)。分類站的位置以鉛垂的虛線標註。相異實線(旁邊標註巴士編號)表示該四台非預備巴士。黑色點狀虛線表示預備巴士。



限制 (Constraints)

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

子任務 (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) 無額外限制。

範例評分程式 (Sample Grader)

此範例評分程式以下列格式讀取輸入：

- 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

此範例評分程式以下列格式輸出你的答案：

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