Overtaking

Hay una vía de un solo carril con único sentido desde el Aeropuerto de Budapest hasta el Hotel Forrás. Esta vía tiene L kilómetros de largo.

En la IOI 2023, $N+1$ buses de transferencia recorrerán esta vía. Los buses están enumerados de 0 a N. El bus i ($0 \le i < N$) está programado para salir del aeropuerto a los $T[i]$ segundos del evento, y puede viajar 1 kilómetros en $W[i]$ segundos. El bus N es un bus reservado y puede viajar 1 kilómetros en X segundos. El tiempo Y en el que saldrá del aeropuerto no ha sido decidido aún.

En general no está permitido adelantar en la vía, pero los buses tienen permitido adelantarse entre sí en **estaciones de servicio**. Hay M ($M > 1$) estaciones de servicio, enumeradas de 0 a $M-1$ en diferentes posiciones de la vía. La estación j ($0 \le j < M$) está ubicada a $S[j]$ kilómetros del aeropuerto a lo largo de la vía. Las estaciones están ordenadas de manera creciente de acuerdo a su distancia desde el aeropuerto, esto es, $S[j] < S[j+1]$ para cada $0 \le j \le M-2$. La primera estación es el aeropuerto y la última es el hotel, es decir, $S[0] = 0$ y $S[M-1] = L$.

Cada bus viaja a su máxima velocidad a menos que alcancen a un bus más lento que viaje delante de él, y en dicho caso están forzados a agruparse y viajar a la velocidad del bus más lento hasta que hayan llegado a la siguiente estación de servicio. Ahí, los buses más rápidos adelantarán a los más lentos.

Formalmente, para cada i y j tales que $0 \le i \le N$ y $0 \le j < M$, el tiempo $t_{i,j}$ (en segundos) cuando el bus i **llega a** la estación de servicio j se define como sigue. Sea $t_{i,0} = T[i]$ para cada $0 \le i < N$, y sea $t_{N,0} = Y$. Para cada j tal que $0 < j < M$:

- Defínase el **tiempo esperado de llegada** (en segundos) del bus i a la estación j, denotada por $e_{i,j}$, como el tiempo en el que el bus i llegaría a la estación j si estuviese viajando a máxima velocidad desde el momento en que llegó a la estación $j-1$. Es decir, sea
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para cada $0 \le i < N$, y
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- El bus i llega a la estación j en el *máximo* de los tiempos esperados de llegada del bus i y de todos los otros buses que llegaron a la estación $j-1$ antes que el bus i. Formalmente, sea $t_{i,j}$ el máximo de los $e_{i,j}$ y de todo $e_{k,j}$ para el que $0 \le k \le N$ y $t_{k,j-1} < t_{i,j-1}$.

Los organizadores de la IOI quieren programar el bus reservado (bus N). Tu tarea es responder Q preguntas de los organizadores, las cuáles son de la siguiente forma: dado el tiempo Y (en segundos) en el que el bus reservado se supone que debe salir del aeropuerto, ¿en que momento debería llegar al hotel?

Detalles de Implementación

Tu tarea es implementar los siguientes procedimientos.

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

- L: la longitud de la vía.
- N: el número de buses no reservados.
- T: un arreglo de longitud N que describe los tiempos en los cuáles los buses no reservados están programados para salir del aeropuerto.
- W: un arreglo de longitud N que describe la máxima velocidad de los buses que no son reservados.
- X: el tiempo que le toma al bus reservado viajar 1 kilómetro.
- M: el número de estaciones de servicio.
- S: un arreglo de longitud M que describe las distances de las estaciones de servicio desde el aeropuerto.
- Este procedimiento es llamado exactamente una vez por cada caso de prueba antes de cualquier llamada a `arrival_time`

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

- Y: el tiempo en el que se supone que el bus reservado (bus N) saldrá del aeropuerto.
- Este procedimiento debe retornar el tiempo en el que el bus reservado debería llegar al hotel.
- Este procedimiento es llamado exactamente Q veces.

Ejemplo

Considera la siguiente secuencia de llamadas:

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

Ignorando al bus 4 (el cuál no ha sido programado aún), la siguiente tabla muestra los tiempos de llegada esperados y reales de los buses no reservados para cada estación de servicio:

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

Los tiempos de llegada a la estación 0 son los tiempos en los cuales los buses están programados para salir del aeropuerto. Esto es, $t_{i,0} = T[i]$ para $0 \le i \le 3$.

Los tiempos de llegada esperados y reales a la estación 1 son calculados de las siguiente forma:

- Los tiempos de llegada esperados en la estación 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: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Los tiempos de llegada a la estación 1:
  - Los buses 1 y 3 llegan a la estación 0 antes que el bus 0, entonces $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - El bus 3 llega a la estación 0 antes que el bus 1, entonces $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Los buses 0, 1 y 3 llegan a las estaciones 0 antes que el bus 2, entonces $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Ningún bus llega a la estación 0 antes que el bus 3, entonces $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

Al bus 4 le toma 10 segundos viajar 1 kilómetro y ahora es programado para salir del aeropuerto a los 0 segundos. En este caso, la siguiente tabla muestra los tiempos de llegada para cada bus. El único cambio con respecto a los tiempos de llegada esperados y reales de los buses no reservados está subrayado.

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

Vemos que el bus 4 llega al hotel a los 60 segundos. Por lo tanto, el procedimiento debe retornar 60

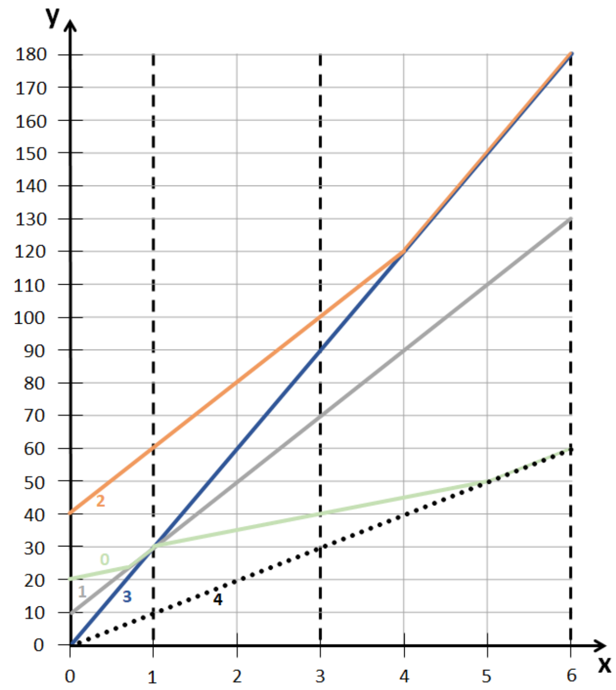
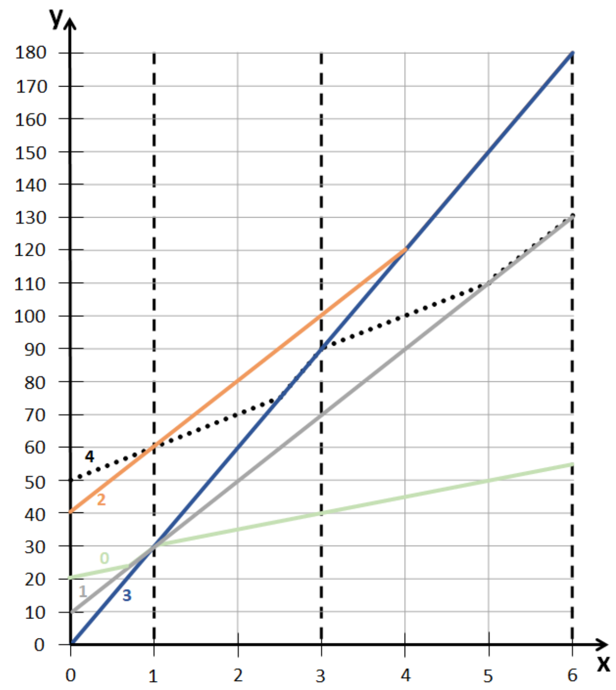
```
arrival_time(50)
```

El bus 4 ahora está programado para salir del aeropuerto a los 50 segundos. En este caso, no hay cambios en los tiempos de llegada de los buses no reservados en comparación con la tabla inicial. Los tiempos de llegada se muestran en la siguiente tabla.

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

El bus 4 adelanta al bus más lento 2 en la estación 1 al llegar al mismo tiempo. Luego, el bus 4 es agrupado con el bus 3 entre la estación 1 y la estación 2, haciendo que el bus 4 llegue a la estación 2 a los 90 segundos en lugar de a los 80 segundos. Despues de salir de la estación 2, el bus 4 es agrupado con el bus 1 hasta que llegan al hotel. El bus 4 llega al hotel a los 130 segundos. Por lo tanto, el procedimiento debe retornar 130.

Podemos graficar el tiempo que le toma a cada bus llegar a cada distancia desde el aeropuerto. El eje x de la gráfica representa la distancia del aeropuerto (en kilómetros) y el eje y de la gráfica representa el tiempo (en segundos). Las líneas verticales discontinuas marcan las posiciones de las estaciones de servicio. Líneas discontinuas diferentes (acompañadas por los índices de los buses) representan a los cuatro buses no reservados. La línea negra punteada representa al bus reservado.

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

Restricciones

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1\,000$
- $0 \leq T[i] \leq 10^{18}$ (para cada i tal que $0 \leq i < N$)
- $1 \leq W[i] \leq 10^9$ (para cada i tal que $0 \leq i < N$)
- $1 \leq X \leq 10^9$
- $2 \leq M \leq 1\,000$
- $0 = S[0] < S[1] < \cdots < S[M-1] = L$
- $1 \leq Q \leq 10^6$
- $0 \leq Y \leq 10^{18}$

Sub-tareas

1. (9 puntos) $N = 1, Q \leq 1\,000$
2. (10 puntos) $M = 2, Q \leq 1\,000$
3. (20 puntos) $N, M, Q \leq 100$
4. (26 puntos) $Q \leq 5\,000$
5. (35 puntos) Sin restricciones adicionaes.

Evaluador de ejemplo

El evaluador de prueba leerá la entrada con el siguiente formato:

- línea 1: $L\ N\ X\ M\ Q$
- línea 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- línea 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- línea 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- línea $5+k$ ($0 \leq k < Q$): Y para la pregunta k

El evaluador de prueba imprimirá tus respuestas con el siguiente formato:

- línea $1+k$ ($0 \leq k < Q$): el valor retornado por `arrival_time` para la pregunta k