Overtaking

Existe uma estrada com uma única faixa de rodagem e um único sentido desde o aeroporto de Budapeste até ao Hotel Forrás. A estrada tem um comprimento de L km.

Durante as IOI 2023, $N+1$ autocarros de transferência percorrem esta estrada. Os autocarros são numerados de 0 até N. O autocarro i ($0 \le i < N$) está previsto sair do aeroporto no $T[i]$-ésimo segundo do evento e consegue percorrer 1 km em $W[i]$ segundos. O autocarro N é um autocarro de reserva que pode viajar 1 km em X segundos. O tempo Y em que deve sair do aeroporto não foi ainda decidido.

Uma ultrapassagem não é permitida na estrada em geral, mas os autocarros podem ultrapassar-se uns aos outros nas **estações (de ultrapassagem)**. Existem M ($M > 1$) estações, numeradas de 0 a $M-1$, em diferentes posições na estrada. A estação j ($0 \le j < M$) está localizada a $S[j]$ km do aeroporto ao longo da estrada. As estações estão ordenadas de forma crescente pela distância ao aeroporto, isto é, $S[j] < S[j+1]$ para cada $0 \le j \le M-2$. A primeira estação é o aeroporto e a última é o hotel, ou seja, $S[0] = 0$ e $S[M-1] = L$.

Cada autocarro viaja à velocidade máxima, a não ser que apanhe um autocarro mais lento a viajar à sua frente na estrada, caso em que fica bloqueado e é obrigado a viajar à velocidade do autocarro mais lento, até chegarem à próxima estação. Aí, os autocarros mais rápidos ultrapassam os autocarros mais lentos.

Formalmente, para cada i e j tal que $0 \le i \le N$ e $0 \le j < M$, o tempo $t_{i,j}$ (em segundos) quando o autocarro i **chega** à estação j é definido da maneira seguinte. Seja $t_{i,0} = T[i]$ para cada $0 \le i < N$ e seja $t_{N,0} = Y$. Para cada j tal que $0 < j < M$:

- Defina-se o **tempo esperado de chegada** (em segundos) do autocarro i na estação j, denotado por $e_{i,j}$, como o tempo em que o autocarro i chegaria à estação j se viajasse sempre à sua velocidade máxima desde a altura em que chegou à estação $j-1$. Isto é:
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$ para cada $0 \le i < N$, e
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- O autocarro i chega à estação j no *máximo* dos tempos esperados de chegada do autocarro i e de todos os outros autocarros que chegaram à estação $j-1$ antes do autocarro i. Formalmente, seja $t_{i,j}$ o máximo entre $e_{i,j}$ e todos os $e_{k,j}$ para os quais $0 \le k \le N$ e $t_{k,j-1} < t_{i,j-1}$.

Os organizadores das IOI querem agendar a saída do autocarro de reserva (o autocarro N). A tua tarefa é responder a Q questões dos organizadores, cada uma no seguinte formato: dado o tempo

Y (em segundos) em que o autocarro de reserva é suposto sair do aeroporto, a que horas chegaria ao hotel?

Detalhes de Implementação

A tua tarefa é implementar as seguintes funções.

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

- L: o comprimento da estrada.
- N: o número de autocarros que não são de reserva.
- T: um array de tamanho N descrevendo os tempos em que está previsto que autocarros que não são de reserva saiam do aeroporto.
- W: um array de tamanho N descrevendo as velocidades máximas dos autocarros que não são de reserva.
- X: o tempo que o autocarro de reserva demora a percorrer 1 km.
- M: o número de estações (de ultrapassagem).
- S: um array de tamanho M descrevendo as distâncias das estações ao aeroporto.
- Esta função será chamada exatamente uma vez para cada caso de teste, antes de qualquer chamada a `arrival_time`.

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

- Y: o tempo em que o autocarro de reserva (autocarro N) é suposto sair do aeroporto.
- Esta função deve devolver o tempo no qual o autocarro de reserva chegaria ao hotel.
- Esta função é chamada exatamente Q vezes.

Exemplos

Considera a seguinte sequência de chamadas:

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

Ignorando o autocarro 4 (que ainda não tem saída agendada), a seguinte tabela mostra os tempos esperados e os tempos reais de chegada para todos os autocarros que não são de reserva em cada uma das estações:

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

Os tempos de chegada à estação 0 são os tempos em que está previsto os autocarros sairem do aeroporto. Isto é $t_{i,0} = T[i]$ para $0 \leq i \leq 3$.

Os tempos esperados e reais de chegada à estação 1 são calculados da seguinte maneira:

- Os tempos esperados de chegada à estação 1:
  - Autocarro 0: $e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - Autocarro 1: $e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - Autocarro 2: $e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - Autocarro 3: $e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- Os tempos reais de chegada à estação 1:
  - Os autocarros 1 e 3 chegam à estação 0 antes do autocarro 0, portanto $t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$.
  - O autocarro 3 chega à estação 0 antes do autocarro 1, portanto $t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$.
  - Os autocarros 0, 1 e 3 chegam à estação 0 antes do autocarro 2, portanto $t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$.
  - Nenhum autocarro chega à estação 0 antes do autocarro 3, portanto $t_{3,1} = \max([e_{3,1}]) = 30$.

```
arrival_time(0)
```

O autocarro 4 leva 10 segundos para percorrer 1 km e está agora agendado para sair no segundo 0. Neste caso, a tabela seguinte mostra os tempos de chegada de cada autocarro. A única alteração no que toca aos tempos esperados e reais dos autocarros que não são de reserva está sublinhada.

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 o autocarro 4 chega ao hotel no segundo 60. Por isso, a função deve devolver 60.

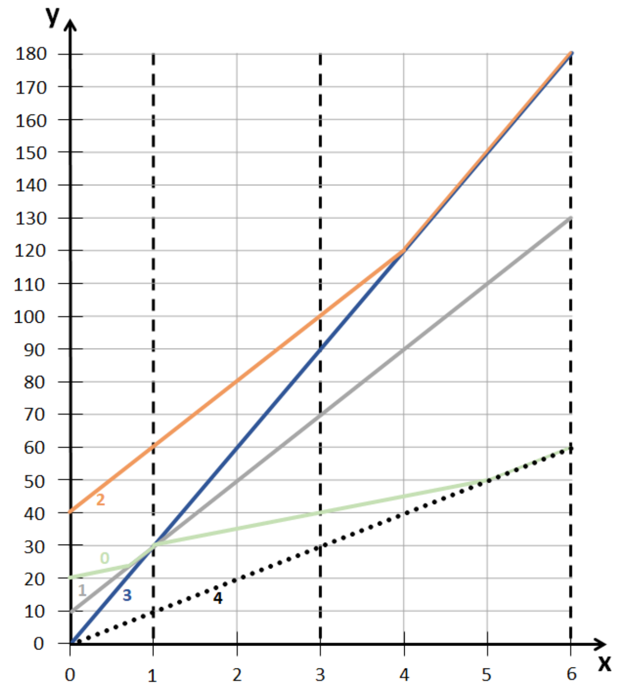
```
arrival_time(50)
```

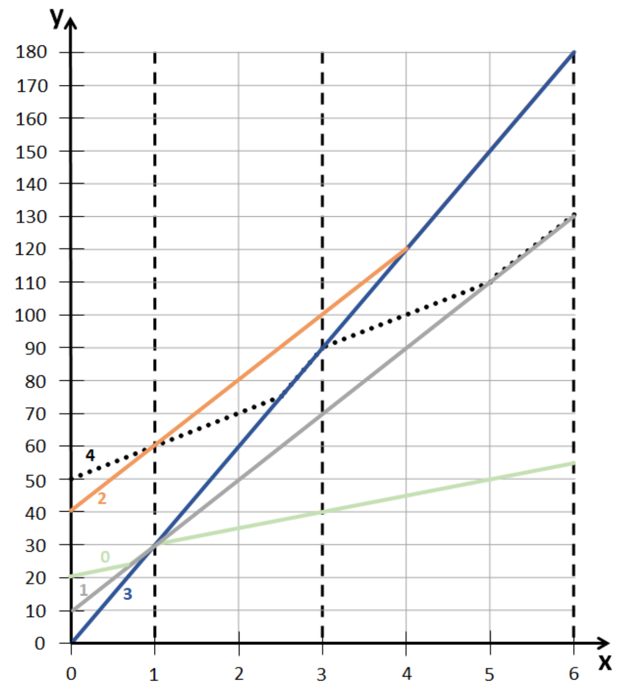
O autocarro 4 está agora agendado para sair no segundo 50. Neste caso, não existem mudaças nos tempos de chegada dos autocarros que não são de reserva quando comparados com a tabela inicial. Os tempos de chegada são mostrados na seguinte tabela.

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

O autocarro 4 ultrapassa o mais lento autocarro 2 na estação 1 porque chegam ao mesmo tempo. A seguir, o autocarro 4 fica bloqueado pelo autocarro 3 entre as estações 1 e 2, fazendo com que o autocarro 4 chegue à estação 2 no segundo 90 em vez do segundo 80. Depois de deixar a estação 2 , o autocarro 4 fica bloqueada pelo autocarro 1 até chegarem ao hotel. O autocarro 4 chega ao hotel no segundo 130. Por isso, a função deve devolver 130.

Podemos fazer um gráfico do tempo que cada autocarro demora a chegar a cada distância do aeroporto. O eixo dos x representa a distância ao aeroporto (em kms) e o eixo dos y representa o tempo (em segundos). As linhas verticais a tracejado marcam as posições das estações. As diferentes linhas contínuas (acompanhadas dos índices dos autocarros) representam os 4 autocarros que não são de reserva. A linha pontilhada (tracejado com pontos) representa o autocarro de reserva.

<table>
<tr><th><code>arrival_time(0)</code></th><th><code>arrival_time(50)</code></th></tr>
<tr><td>


</td><td>


</td></tr>
</table>

Restrições

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

Subtarefas

1. (9 pontos) $N = 1, Q \leq 1\,000$
2. (10 pontos) $M = 2, Q \leq 1\,000$
3. (20 pontos) $N, M, Q \leq 100$
4. (26 pontos) $Q \leq 5\,000$
5. (35 pontos) Nenhuma restrição adicional

Avaliador Exemplo

O avaliador exemplo lê o input no seguinte formato:

- linha 1: $L\ N\ X\ M\ Q$
- linha 2: $T[0]\ T[1]\ \ldots\ T[N-1]$
- linha 3: $W[0]\ W[1]\ \ldots\ W[N-1]$
- linha 4: $S[0]\ S[1]\ \ldots\ S[M-1]$
- linha $5+k$ ($0 \le k < Q$): Y para a pergunta k

O avaliador exemplo escreve as tuas respostas no seguinte formato:

- linha $1+k$ ($0 \le k < Q$): o valor devolvido por `arrival_time` na questão k