バスの追い越し（Overtaking）

ブダペスト空港から Hotel Forrás までの間には，一方通行で一車線の道路がある．この道路の長さは L キロメートルである．

IOI 2023 のイベントの間，この道路を $N+1$ 台のバスが通過する．バスには 0 から N までの番号が付けられている． バス i（$0 \le i < N$）はイベントが始まってから $T[i]$ 秒目に空港を出発する予定であり，$W[i]$ 秒で 1 キロメートルを進むことができる． バス N は予備のバスであり，X 秒で 1 キロメートルを進むことができる． この予備のバスが空港を出発する時刻 Y はまだ決められていない．

この道路では基本的に追い越しは許可されていないが，バスは **整列所** で互いに追い越しをすることができる． この道路には M 個（$M > 1$）の整列所があり，0 から $M-1$ までの番号が付けられている． 整列所の場所は相異なっており，整列所 j（$0 \le j < M$）は空港から道路沿いに $S[j]$ キロメートルの位置にある． 整列所は空港からの距離が近い順に番号付けされている．すなわち，$0 \le j \le M-2$ なる各 j について $S[j] < S[j+1]$ である． 最初の整列所は空港にあり，最後の整列所はホテルにある． すなわち，$S[0] = 0$ かつ $S[M-1] = L$ である．

各バスは，前方を走るより遅いバスに追いつかない限りは，最高速度で走る．追いついた場合，次の整列所に到着するまで，そのより遅いバスと同じ速度で一緒に走らなければならない．整列所に到着すると，そこで速い方のバスは遅い方のバスを追い越す．

形式的には，$0 \le i \le N$ かつ $0 \le j < M$ であるようなそれぞれの組 i, j について，バス i が整列所 j に **到着する** 時刻 $t_{i,j}$（単位は秒）は以下のように定義される． まず，$t_{i,0} = T[i]$（$0 \le i < N$），$t_{N,0} = Y$ とする． $0 < j < M$ であるような各 j について，

- バス i が整列所 $j-1$ に到着してから常に最高速度で走れた場合の整列所 j への到着時刻を，バス i の整列所 j への **予想到着時刻** $e_{i,j}$（単位は秒）と定義しよう．すなわち，以下のように定義する．
  - $e_{i,j} = t_{i,j-1} + W[i] \cdot (S[j] - S[j-1])$（$0 \le i < N$）であり，かつ
  - $e_{N,j} = t_{N,j-1} + X \cdot (S[j] - S[j-1])$.
- バス i の整列所 j への到着時刻は，バス i の予想到着時刻とバス i よりも早く整列所 $j-1$ に到着するバスの予想到着時刻の *最大値* である．形式的には，$t_{i,j}$ は， $e_{i,j}$ と， $0 \le k \le N$ かつ $t_{k,j-1} < t_{i,j-1}$ なるすべての k についての $e_{k,j}$ の最大値である．

IOI の運営は予備のバス（バス N）の運行計画を立てようとしている． 「バス N が空港を出発する時刻が Y（単位は秒）であるとき，このバスがホテルに到着するのはいつか」という形式の質問が Q 回与えられる．これらの質問に答えよ．

実装の詳細

以下の関数を実装せよ．

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

- L：道路の長さ．
- N：予備でないバスの台数．
- T：予備でないバスが空港を出発する時刻を表す長さ N の配列．
- W：予備でないバスの最高速度を表す長さ N の配列．
- X：予備のバスが 1 キロメートルを走るのに要する秒数．
- M：整列所の個数．
- S：整列所の空港からの距離を表す，長さ M の配列．
- この関数は，各テストケースについて，`arrival_time` が呼び出されるよりも前にちょうど 1 回呼び出される．

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

- Y：予備のバス（バス N）が空港を出発する時刻．
- この関数は予備のバスがホテルに到着する時刻を返す必要がある．
- この関数はちょうど Q 回呼び出される．

例

以下のような連続した呼び出しを考えよう：

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

まだ運行計画が決まっていないバス 4 を除いて考えると，予備でないバスの各整列所への予想到着時刻と実際の到着時刻は以下の表の通りである．

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

整列所 0 への到着時刻は，各バスの空港を出発する時刻に一致する．すなわち，$0 \le i \le 3$ について，$t_{i,0} = T[i]$ である．

各バスの整列所 1 への予想到着時刻と実際の到着時刻は以下の通りに計算される：

- 整列所 1 への予想到着時刻：
  - バス 0：$e_{0,1} = t_{0,0} + W[0] \cdot (S[1] - S[0]) = 20 + 5 \cdot 1 = 25$.
  - バス 1：$e_{1,1} = t_{1,0} + W[1] \cdot (S[1] - S[0]) = 10 + 20 \cdot 1 = 30$.
  - バス 2：$e_{2,1} = t_{2,0} + W[2] \cdot (S[1] - S[0]) = 40 + 20 \cdot 1 = 60$.
  - バス 3：$e_{3,1} = t_{3,0} + W[3] \cdot (S[1] - S[0]) = 0 + 30 \cdot 1 = 30$.
- 整列所 1 への実際の到着時刻：
  - バス 0 について，バス 1,3 がより早く整列所 0 に到着するため，$t_{0,1} = \max([e_{0,1}, e_{1,1}, e_{3,1}]) = 30$ となる．
  - バス 1 について，バス 3 がより早く整列所 0 に到着するため，$t_{1,1} = \max([e_{1,1}, e_{3,1}]) = 30$ となる．
  - バス 2 について，バス 0,1,3 がより早く整列所 0 に到着するため，$t_{2,1} = \max([e_{0,1}, e_{1,1}, e_{2,1}, e_{3,1}]) = 60$ となる．
  - バス 3 よりも早く整列所 0 に到着するバスはないため，$t_{3,1} = \max([e_{3,1}]) = 30$ となる．

```
arrival_time(0)
```

バス 4 は 1 キロメートルを走るのに 10 秒を要し，0 秒目に空港を出発することになった．このとき，各バスの到着時刻は以下の表のようになる．下線が引かれた 1 箇所について，予備でないバスの予想到着時刻と実際の到着時刻が変わっている．

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

バス 4 はホテルに 60 秒目に到着することがわかる． したがって，この関数は 60 を返す必要がある．

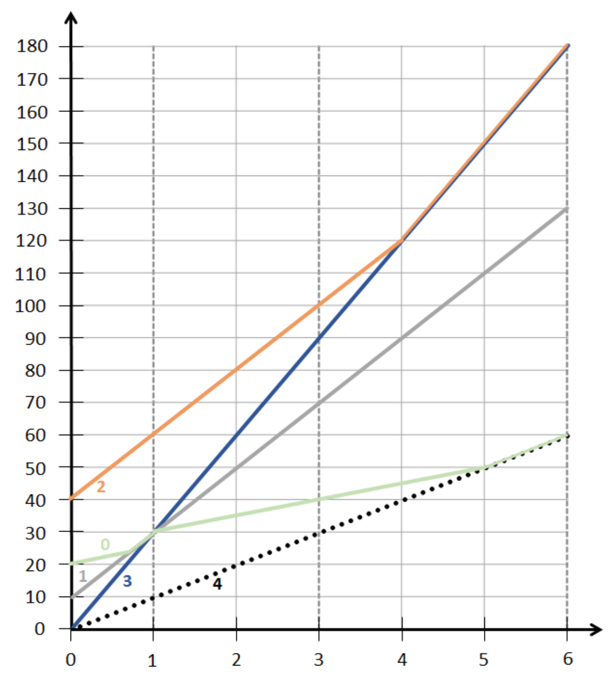
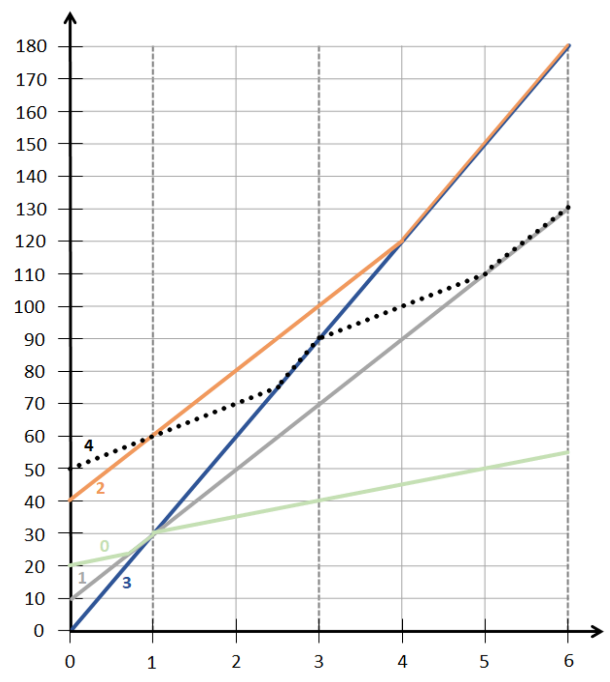
```
arrival_time(50)
```

バス 4 は 50 秒目に空港を出発することになった．このとき，予備でないバスの予想到着時刻と実際の到着時刻は元の表から変わらない．到着時刻は以下の表の通りである．

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

バス 4 はより遅いバス 2 と同時に整列所 1 に到着し，これを追い越す． 次に，バス 4 は整列所 1 と整列所 2 の間でバス 3 と一緒になり，これによりバス 4 の整列所 2 への到着時刻は 80 秒目ではなく 90 秒目となる． 整列所 2 を出発した後，ホテルに到着するまでの間に，バス 4 はバス 1 と一緒になる． バス 4 がホテルに到着するのは 130 秒目になる． したがって，この関数は 130 を返す必要がある．

それぞれのバスが空港から各距離に至るまでに要する時間をグラフにした． x 軸は空港からの距離（単位はキロメートル）を表しており，y 軸は時刻（単位は秒）を表している． 縦の破線は整列所の位置を示している． バスの番号が振られているそれぞれの実線は，4 台の予備でないバスを表している． 黒の点線は予備のバスを表している．

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

制約

- $1 \leq L \leq 10^9$
- $1 \leq N \leq 1\,000$

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

小課題

1. (9 点) $N = 1, Q \leq 1\,000$
2. (10 点) $M = 2, Q \leq 1\,000$
3. (20 点) $N, M, Q \leq 100$
4. (26 点) $Q \leq 5\,000$
5. (35 点) 追加の制約はない．

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む：

- 1 行目: $L\ N\ X\ M\ Q$
- 2 行目: $T[0]\ T[1]\ \ldots\ T[N-1]$
- 3 行目: $W[0]\ W[1]\ \ldots\ W[N-1]$
- 4 行目: $S[0]\ S[1]\ \ldots\ S[M-1]$
- $5 + k$ ($0 \leq k < Q$) 行目: k 回目の質問における Y

採点プログラムのサンプルは以下の形式であなたの答えを出力する：

- $1 + k$ ($0 \leq k < Q$) 行目: k 回目の質問に対する `arrival_time` の戻り値