



Baker

JOI Bakery is a bakery famous for its mouth-watering croissants. JOI Bakery has N bakers numbered from 1 to N . Baker i ($1 \leq i \leq N$) takes i minutes to make one croissant. One baker cannot make multiple croissants at the same time.

Today, M customers numbered from 1 to M are scheduled to visit JOI Bakery, and each customer plans to order one croissant. Customer j ($1 \leq j \leq M$) will order a croissant at time T_j , where time t denotes the time t minutes from now. However, a customer who cannot receive their croissant within L minutes of ordering will give up and leave the shop. In other words, to fulfill the order of customer j ($1 \leq j \leq M$), the croissant must be finished by time $T_j + L$ (including exactly at time $T_j + L$).

Manager K, who manages JOI Bakery, plans to have exactly one baker work today and is considering which baker to send and at what time. Since bakers focus intensely on bread-making during their shift, **they ignore all orders placed after their start time (not including exactly at the start time)**. That is, a baker starting work at time t cannot fulfill orders from customers j ($1 \leq j \leq M$) such that $T_j > t$.

Manager K is currently considering Q work plans. The q -th plan ($1 \leq q \leq Q$) is to have baker A_q start work at time B_q . To help with the decision, for each of the Q plans, he wants to know the maximum number of customers whose orders can be fulfilled if that plan is executed. Note that the time it takes for a baker to start making a croissant after arriving, and the time it takes to start making a new croissant after finishing one, can be ignored.

Given information about the customers visiting JOI Bakery and the work plans, create a program to find the maximum number of customers whose orders can be fulfilled for each plan.

Input

Read the following data from the standard input.

```
 $N$   $M$   $L$   $Q$   
 $T_1$   $T_2$   $\cdots$   $T_M$   
 $A_1$   $B_1$   
 $A_2$   $B_2$   
 $\vdots$   
 $A_Q$   $B_Q$ 
```



Output

Write Q lines to the standard output. The q -th line ($1 \leq q \leq Q$) of the output should contain an integer representing the maximum number of customers whose orders can be fulfilled in the q -th work plan.

Constraints

- $1 \leq N \leq 4 \times 10^{12}$.
- $1 \leq M \leq 2\,000\,000$.
- $1 \leq L \leq 2 \times 10^{12}$.
- $1 \leq Q \leq 400\,000$.
- $0 \leq T_j \leq 2 \times 10^{12}$ ($1 \leq j \leq M$).
- $T_j \leq T_{j+1}$ ($1 \leq j \leq M - 1$).
- $1 \leq A_q \leq N$ ($1 \leq q \leq Q$).
- $0 \leq B_q \leq 4 \times 10^{12}$ ($1 \leq q \leq Q$).
- Given values are all integers.

Subtasks

1. (8 points) $M \leq 10$, $Q \leq 100\,000$.
2. (12 points) $M \leq 500$, $Q \leq 100\,000$.
3. (30 points) $T_M \leq B_q < T_1 + L$ ($1 \leq q \leq Q$).
4. (10 points) $T_M \leq B_q$ ($1 \leq q \leq Q$).
5. (22 points) $M \leq 500\,000$, $Q \leq 100\,000$.
6. (18 points) No additional constraints.



Sample Input and Output

Sample Input 1	Sample Output 1
4 4 6 4	3
0 2 3 8	2
2 3	2
1 6	0
3 3	
4 7	

Regarding the 1st work plan, baker 2 starting at time 3 can fulfill the orders of 3 customers 1, 2, and 3, for example, as follows:

- First, to fulfill customer 1's order, start making a croissant at time 3 and finish it 2 minutes later at time 5. (This satisfies the condition to finish by time $T_1 + L = 0 + 6 = 6$.)
- Next, to fulfill customer 2's order, start making a croissant at time 5 and finish it 2 minutes later at time 7. (This satisfies the condition to finish by time $T_2 + L = 2 + 6 = 8$.)
- Finally, to fulfill customer 3's order, start making a croissant at time 7 and finish it 2 minutes later at time 9. (This satisfies the condition to finish by time $T_3 + L = 3 + 6 = 9$.)

Customer 4's order is ignored because it arrives after the baker's start time, so it cannot be fulfilled. Thus, a maximum of 3 customers' orders can be fulfilled, so the 1st line outputs 3.

Regarding the 2nd work plan, baker 1 starting at time 6 can fulfill the orders of 2 customers 2 and 3, for example, as follows:

- First, to fulfill customer 3's order, start making a croissant at time 6 and finish it 1 minute later at time 7. (This satisfies the condition to finish by time $T_3 + L = 3 + 6 = 9$.)
- Next, to fulfill customer 2's order, start making a croissant at time 7 and finish it 1 minute later at time 8. (This satisfies the condition to finish by time $T_2 + L = 2 + 6 = 8$.)

Customer 4's order is ignored because it arrives after the start time. Also, the order of customer 1, which needs to be finished by time 6, cannot be fulfilled. Thus, a maximum of 2 customers' orders can be fulfilled, so the 2nd line outputs 2.



Regarding the 3rd work plan, baker 3 starting at time 3 can fulfill orders for customers 1 and 3, or customers 2 and 3, but cannot fulfill all orders for customers 1, 2, and 3. They also cannot fulfill the order for customer 4 which arrives after the start time. Thus, a maximum of 2 customers' orders can be fulfilled, so the 3rd line outputs 2.

Regarding the 4th work plan, baker 4 starting at time 7 cannot fulfill any customer's order. Thus, the 4th line outputs 0.

This sample input satisfies the constraints of Subtasks 1, 2, 5, and 6.

Sample Input 2	Sample Output 2
20 5 12 4	5
1 2 4 8 10	4
1 12	3
3 10	0
3 11	
15 10	

This sample input satisfies the constraints of all the subtasks.

Sample Input 3	Sample Output 3
100000 6 272273 10	2
5 9 209 8128 17202 50102	2
164 9	4
11 24	3
835 9267	1
2 256	2
2 314156	5
18475 142	0
1826 18978	3
44757 1	2
4 1646	
218 44	

This sample input satisfies the constraints of Subtasks 1, 2, 5, and 6.