Triple Attack

Input file: `standard input`
Output file: `standard output`
Time limit: 3 seconds
Memory limit: 1024 megabytes

Zeus is analyzing a replay of the fight to understand his opponent's attack patterns. The opponent has a special ability: if they land three attacks on a target within a time frame of z, their third attack becomes a powerful, boosted attack.

To outplay his opponent, Zeus should not let his opponent trigger their boosted attack. Let $Y = \{y_1, y_2, \ldots, y_m\}$ be the multiset of m timestamps, where each y_i represents the time when the opponent's attack landed. We call Y to be **safe** if for every three timestamps $\{y_i, y_j, y_k\}$ such that $1 \le i < j < k \le m$, it holds that $\max(y_i, y_j, y_k) - \min(y_i, y_j, y_k) > z$, where z is the duration of the time frame that is given to you as an input.

Zeus has a log of n timestamps, $x_1, x_2, \ldots, x_n$, representing when the opponent's attacks landed. The timestamps are **sorted in nondecreasing order** of occurrence. In other words, $x_i \le x_{i+1}$ for all $1 \le i < n$.

Zeus has q intervals of interest, denoted as two integers $1 \le l \le r \le n$. For each interval, Zeus wants to find the maximum number of attacks among $[x_l, x_{l+1}, \ldots, x_r]$ that he could have let through: In other words, Zeus wants to find a maximum size subset of the multiset $\{x_l, x_{l+1}, \ldots, x_r\}$ such that the subset is **safe**.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 20\,000$). The description of the test cases follows.

The first line contains two integers n and z ($1 \le n \le 250\,000$, $1 \le z \le 10^9$).

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le 10^9$) — the timestamps of the opponent's attacks. It is guaranteed that the array x is sorted, i.e., $x_i \le x_{i+1}$ for all $1 \le i < n$.

The third line contains a single integer q ($1 \le q \le 250\,000$).

The next q lines each contain two integers l and r ($1 \le l \le r \le n$) — the endpoints of the interval.

It is guaranteed that the sum of n over all test cases does not exceed $250\,000$.

It is guaranteed that the sum of q over all test cases does not exceed $250\,000$.

Output

For each of the q queries, print a single integer — the maximum size of a safe subset of attack timestamps in the given interval.

Example

standard input	standard output
3 6 10 1 5 7 8 11 12 6 1 6 1 5 2 6 1 4 2 5 3 6 6 1 1 1 1 3 3 3 2 3 3 1 6 12 15 4 5 15 24 27 32 36 39 40 46 48 48 20 1 12 1 11 6 10 1 8 8 12 11 12 2 9 3 8 7 8 7 10 4 8 9 12 9 10 2 12 1 5 3 12 4 8 3 7 7 12 10 11	3 2 2 2 2 2 1 4 6 6 2 4 2 2 5 3 2 2 2 2 2 6 4 5 2 3 2 2

Note

In the first query of the first test case, we consider the timestamps $\{1, 5, 7, 8, 11, 12\}$ with $z = 10$. The subset $\{1, 5, 12\}$ is safe because its only triplet satisfies $12 - 1 = 11 > 10$. It's impossible to form a safe subset of size 4, hence the answer to this query is 3.

In the first query of the second test case, we consider the timestamps $\{1\}$ with $z = 1$. The entire set $\{1\}$ is safe because there are no triplets. Hence the answer to this query is 1.

In the second query of the second test case, we consider the timestamps $\{1, 1, 1, 3, 3, 3\}$ with $z = 1$.

The subset $S = \{1, 1, 3, 3\}$ is safe because:

- For the triple $(i, j, k) = (1, 2, 3)$, $\max(1, 1, 3) - \min(1, 1, 3) = 2 > 1$.

- For the triple $(i, j, k) = (1, 2, 4)$, $\max(1, 1, 3) - \min(1, 1, 3) = 2 > 1$.
- For the triple $(i, j, k) = (1, 3, 4)$, $\max(1, 3, 3) - \min(1, 3, 3) = 2 > 1$.
- For the triple $(i, j, k) = (2, 3, 4)$, $\max(1, 3, 3) - \min(1, 3, 3) = 2 > 1$.

It's impossible to form a safe subset of size 5, hence the answer to this query is 4.