Triple Removal

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Tired of supporting ranged carries, Keria is now creating a data structure problem about supporting range queries.

For an array $b = [b_1, b_2, \ldots, b_m]$ of length m where $b_i = 0$ or $b_i = 1$, consider the following **triple removal** operation:

1. Choose three indices $1 \le i < j < k \le m$ such that the elements at these positions are identical ($b_i = b_j = b_k$).
2. Remove these three elements from the array. The cost of this operation is defined as $\min(k-j, j-i)$. After the removal, the remaining parts of the array are concatenated, and their indices are relabeled.

We want to make the array b empty using the **triple removal** operation. Hence, we define the *total cost* of an array as the minimum possible sum of the costs of **triple removal** operations required to make the array empty. If it is impossible to make the array empty, the cost is defined to be -1.

Keria wants to test his data structure. For this, you must answer q independent queries. Initially, you are given an array $a = [a_1, a_2, \ldots, a_n]$ of length n where $a_i = 0$ or $a_i = 1$. For each query, you are given a range $1 \le l \le r \le n$ and must find the cost for the array $[a_l, a_{l+1}, \ldots, a_r]$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and q ($1 \le n, q \le 250\,000$) — the length of the array and the number of queries.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($a_i = 0$ or $a_i = 1$) — the elements of the array.

Then q lines follow. The i-th of them contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the range of the subarray for the i-th query.

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 test case, output q lines. The i-th line should contain a single integer representing the answer to the i-th query.

Example

standard input	standard output
2 12 4 0 0 1 1 0 1 0 1 0 1 1 0 1 12 2 7 5 10 6 11 6 3 0 0 0 1 1 1 1 3 4 6 1 6	4 2 3 -1 1 1 2

Note

Explanation of the first test case, first query (1 12):

The subarray is $[0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0]$. There are six 0s and six 1s. A possible optimal sequence of operations is:

1. Remove the three 1s at indices 3, 4, 6. The cost is $\min(6-4, 4-3) = \min(2, 1) = 1$. The array becomes $[0, 0, 0, 0, 1, 0, 1, 1, 0]$.
2. Remove the three 0s at indices 1, 2, 3. The cost is $\min(3-2, 2-1) = \min(1, 1) = 1$. The array becomes $[0, 1, 0, 1, 1, 0]$.
3. Remove the three 1s at indices 2, 4, 5. The cost is $\min(5-4, 4-2) = \min(1, 2) = 1$. The array becomes $[0, 0, 0]$.
4. Remove the three 0s at indices 1, 2, 3. The cost is $\min(3-2, 2-1) = \min(1, 1) = 1$. The array is now empty.

The total cost is $1 + 1 + 1 + 1 = 4$.