

Best Trades Again

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes

This is another version of Problem A, All the Trades Are the Best. In this version, each working period has its own transaction limit and lies entirely within one day. The working periods are not required to be generated randomly.

You are a trader at an investment company. One day, you discover that the stock you're responsible for trading follows an exactly repeating daily pattern: the day is divided into n equal time intervals, and the stock's price during the i -th interval is always a_i . As the n -th interval of one day concludes, the first interval of the next day immediately follows, with the price shifting from a_n to a_1 . This is a once-in-a-lifetime opportunity — if you can just buy and sell according to this pattern you've already figured out, during your working hours, there's a guaranteed way to profit!

You are given q independent working periods. The i -th period starts at slot l_i and ends at slot r_i , including both endpoints, where $l_i \leq r_i$. The slots in chronological order are $l_i, l_i + 1, \dots, r_i$.

In the i -th working period, you may complete at most k_i transactions. Each transaction consists of buying one share and selling it in a later time slot, with profit equal to the sale price minus the purchase price. You may hold at most one share at a time, so you must sell your current share before buying another. All purchases and sales must take place within that working period and follow the chronological order above. You may also make no transactions, earning a profit of 0.

For each working period, independently find the maximum total profit you can earn.

Input

The first line contains two integers n, q ($1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$): the number of time slots in a day and the number of working periods, respectively.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$), where a_i is the stock price in the i -th time slot of each day.

Each of the next q lines contains three integers l_i, r_i, k_i ($1 \leq l_i \leq r_i \leq n$, $0 \leq k_i \leq \lfloor (r_i - l_i + 1)/2 \rfloor$), describing the i -th working period and its transaction limit.

There is no randomness guarantee for the working periods or their transaction limits.

Output

Output q lines. The i -th line should contain one integer: the maximum profit you can earn during the i -th working period with at most k_i transactions.

Example

standard input	standard output
6 7	8
3 1 4 1 5 9	11
1 6 1	11
1 6 2	4
1 6 3	7
2 5 1	0
2 5 2	0
4 4 0	
1 6 0	

Note

In the first query, buy in slot 2 and sell in slot 6, earning $9 - 1 = 8$.

In the second query, buy in slot 2, sell in slot 3, buy again in slot 4, and sell in slot 6. The total profit is $(4 - 1) + (9 - 1) = 11$. Allowing a third transaction in the third query does not increase the maximum profit.

In the sixth and seventh queries, $k_i = 0$, so no transactions are allowed and the answer is 0.