DIVIDENDS

In short order, you write an efficient algorithm for the Enemy-omatic, and the alpha release is a huge hit. The app spreads like wildfire - turns out many have the issue of irrationally disliking others. However, this demand overloads your servers, and it's time to start finding some funding.

The Venture Capital (VC) convention consists of N booths. While VCs always sounded intimidating in college, you've learned the secret - VC booths essentially just hand out free money. The i'th booth at the convention gives $\$f_i$ to each funding-seeker (**every time they visit the booth**), no questions asked.

When entering the convention, you **start at the first booth**. Each minute, you can speak to one VC (and receive $\$f_i$ for visiting booth i). Since the booths are all laid out in a row, after speaking to VC i, the next booth you go to must be number $i-1$, i, or $i+1$.

You're not yet sure about how long you'll have to stay at the convention, but know it'll be one of Q possible amounts of time. Thus, for each j, determine the **maximum amount of funding you can generate if you stay at the convention for t_j minutes**.

Input

The first line contains two space-separated integers, N and Q.

The second line contains the amounts each VC booth gives upon a single visit as N space-separated integers, $f_1, \cdots, f_N$. These will satisfy $1 \le f_i \le 10^5$, for $1 \le i \le N$.

The third line contains the possible number of minutes you can visit the convention as Q space-separated integers, $t_1, \cdots, t_Q$. These will satisfy $1 \le t_j \le 10^8$, for $1 \le j \le Q$.

Output

For query j, the answer is the maximum amount of funding you can accumulate in t_j minutes at the convention. Find the sum of the answers over all Q queries.

Constraints

In all test cases, $1 \le N, Q \le 2 \cdot 10^5$. Beyond the sample input, the tests are divided into batches with additional constraints. Time limits below are for C/C++; Ocaml gets 2x, Java 3x, and Python 10x.

- 21 points satisfy $Q = 1$ and $N \cdot t_1 \le 10^5$. TL: 100ms.
- 30 points worth satisfy $1 \le N \cdot Q \le 10^5$. TL: 150ms.
- 49 points worth satisfy no further constraints. TL: 500ms.
- 10 points (110 total; extra credit) are given for the extension found at problem DIVIDENDSHARD: In the modification, VC's will not fund you two minutes in a row. Thus, after each minute you can choose to move left a booth or to right a booth, but **cannot stay still**. N will always be at least 2.

Sample explanation

In the first minute, you always visit booth 1 (which is where you start), so the answer to query 1 is 1. With 4 minutes, it is optimal to go to the second booth at the second minute, and stay there for three

minutes - so the answer is 13. With 100 minutes, the optimal strategy is to beeline for booth 5 (which gives the most funding by far) and just stay there; giving answer 873. The sum of these is 887.

View submissions (https://cs124.seas.harvard.edu/problem/DIVIDENDS/code-submission)

Test cases

Input	Output	Points	Timeout

`5 3`
`1 4 3 1 9`
`1 4 100`
	887	0	100 ms

Hidden	*Hidden*	7	100 ms
Hidden	*Hidden*	7	100 ms
Hidden	*Hidden*	7	100 ms
Hidden	*Hidden*	10	150 ms
Hidden	*Hidden*	10	150 ms
Hidden	*Hidden*	10	150 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms
Hidden	*Hidden*	7	500 ms

Download (https://cs124.seas.harvard.edu/problem/DIVIDENDS/test-cases)

Inspired by the "Ultra Cool Programming Contest Control Centre" by Sonny Chan.
Modified for CS 124 by Neal Wu (https://github.com/nealwu), with design help from Martin Camacho.
Further refined by Nikhil Benesch (https://github.com/benesch).