# Sub Array Sum Problem

Time limit: *2000 ms*
Memory limit: *512 MB*

Given an 1D array the sub array sum problem asks for the sum of some subarrays (also called intervals). In 2D the same problem can be modelled as a 2D matrix and sub matrix sums, in 3D we have cubes. We can generalize this problem for every dimension.

Given a $d$-dimensional array $a$ of integers answer queries of getting the sum of $d$-dimensional subarray.

## Standard input

The first line contains one integer $d$ $(1 \leq d \leq 5)$ which represents the number of dimensions of the array.

The second line contains $d$ positive integers which represent a vector $dim$, each integer denoting the length of a dimension. The total size of the array is less than or equal to $10^5$, formally $1 \leq \prod_{i=1}^{d} dim_i \leq 10^5$.

The third line contains $\prod_{i=1}^{d} dim_i$ integers, the values of the array coming in lexicographical order of their indices.

The fourth line contains one integer $q$ $(1 \leq q \leq 10^5)$ the number of queries.

Each query consists of two lines:

- First line of the query is a vector $l$ of positive integers.
- Second line of the query is a vector $r$ of positive integers.

Where:

- $l$ is a $d$-tuple representing the coordinates of the lowest lexicographical index of the $d$-hyper rectangle.
- $r$ is a $d$-tuple representing the coordinates of the highest lexicographical index of the $d$-hyper rectangle.
- Both vectors $l$ and $r$ are $1$-based coordinates and $1 \leq l_i \leq r_i \leq dim_i$ , $\forall i \mid 1 \leq i \leq d$

## Standard output

For every query print the answer in a line , it guarantees the answer fit in a 32bit-signed integer.

## Constraints and notes

- $1 \leq d \leq 5$
- $1 \leq \prod_{i=1}^{d} dim_i \leq 10^5$
- The elements of $a$ fit in $32$-bit signed integers
- $1 \leq q \leq 10^5$

| Input | Output | Explanation |
|---|---|---|
| 1<br>5<br>1 2 3 4 5<br>3<br>1<br>1<br>1<br>5<br>3<br>5 | 1<br>15<br>12 | In the first test case we have an 1D array.<br>For the first query we have 1.<br>For the second query we have 1+2+3+4+5=15.<br>For the third query we have 3+4+5=12. |
| 2<br>3 3<br>1 2 3 4 5 6 7 8 9<br>2<br>1 1<br>1 1<br>1 1<br>2 2 | 1<br>12 | In the second test case we have a 2D array.<br>For the first query we have 1.<br>For the second query we have 1+2+4+5=12. |