Programming Refresher Workshop

Session 3 Exercises

Exercise 8 (ex8): Efficient Adding

Given a large number of integer values stored in an array, if we are asked to add up different portions of the values over and over again, it would be inefficient to repeatedly do the additional. Your task for this exercise is to think of a way to preprocess those integer values so that you can efficiently answer queries about the sum of the values in the given range.

For example, given the following array of values,

25	66	21	89	12	78	26	44	19	93	72	88	64
0	1	2	3	4	5	6	7	8	9	10	11	12

If you are asked to find the sum of values between index 1 and 10 inclusive, you should not add all the values 66+21+…+93+72 to get the total 520.

If you are asked to find the sum of values between index 0 and 12, you should immediately give the total 697 instead of adding up all the values in the range.

You may assume that you will be asked to find the sum of values in a contiguous portion of the array, i.e., from index *i* through index *j*, where $i \leq j$.

You may also assume that there are at most 50,000 values in the array.

Input

The first line of input contains an integer N which indicates all the integer values in the given array. This is followed by N lines each with one integer.

The next line of input contains an integer Q which indicates the number of queries. This is followed by Q lines each with a starting index and ending index to indicate the range of values to be added up.

Output

For each of the Q queries, output a single integer followed by a newline character.

Sample input

```
13
25
66
21
89
12
78
26
44
19
93
72
88
64
2
1 10
0 12
```

Sample output

```
520
697
```

Algorithm template

Input

How to accept all the input

Processing

How to store the values into the array?

What is the preprocess?

How to answer the queries?

Output

How to output the result?