A + B Queries

The Quechuas welcome you to IOI 2025 with a special gift: two arrays, A and B, each of length N. The elements in both arrays are indexed from 0 to $N - 1$.

To ensure that you are taking good care of their gift, they will ask you Q questions, one at a time. Each question consists of two indices, i and j, and asks: What is the sum of $A[i]$ and $B[j]$?

Implementation Details

The first procedure you should implement is:

```
void initialize(std::vector<int> A, std::vector<int> B)
```

- A, B: two arrays of length N, the gift of the Quechuas.
- This procedure is called exactly once for each testcase, before any calls to `answer_question`.

The second procedure you should implement is:

```
int answer_question(int i, int j)
```

- i, j: integers describing a question.
- This procedure is called Q times.

This procedure should return the sum of $A[i]$ and $B[j]$.

Constraints

- $1 \leq N \leq 200\,000$
- $0 \leq A[k], B[k] \leq 10^9$ for each k such that $0 \leq k < N$.
- $1 \leq Q \leq 200\,000$
- $0 \leq i, j < N$ in each question.

Subtasks

Subtask	Score	Additional Constraints
1	25	All elements in array A are equal and all elements in array B are equal.
2	35	$N \leq 1000$
3	40	No additional constraints.

Example

Consider the following call:

```
initialize([2, 1, 3], [0, 7, 8])
```

In this case $N = 3$ and the two arrays gifted to you are $A = [2, 1, 3]$ and $B = [0, 7, 8]$.

Now consider the following call:

```
answer_question(0, 1)
```

This call should return the sum of $A[0] = 2$ and $B[1] = 7$, which is 9.

Consider the following call:

```
answer_question(2, 2)
```

This call should return $A[2] + B[2] = 3 + 8 = 11$.

Sample Grader

Input format:

```
N
A[0] A[1] ... A[N-1]
B[0] B[1] ... B[N-1]
Q
i[0] j[0]
i[1] j[1]
...
i[Q-1] j[Q-1]
```

Here, $i[k]$ and $j[k]$ ($0 \leq k < Q$) specify the parameters for each call to `answer_question`.

Output Format:

```
S[0]
S[1]
...
S[Q-1]
```

Here, $S[k]$ ($0 \le k < Q$) is the integer returned by the call `answer_question(i[k], j[k])`.