

## S2-B. End of The Line

Time limit	2 s
Memory limit	256 MB

### Description

There are  $N$  people in front of a ticket counter lining up for Coco's concert. Surely, each of them wants to have a perfect place to sit. For that reason, Mr. Dengklek, one of the staffs, has received many complaints about the ordering of the seats. Mr. Dengklek wants to give them a chance to arrange their seats.

It is known that each person has their own preferences in determining which seat suits them better. Initially, the  $i$ -th person that has preference  $C_i$  is located in row  $A_i$ . When  $C_i = S$ , it means that the  $i$ -th person wants the row where he sits to be as close as possible to the stage (i.e. it is preferred for  $A_i$  to be as small as possible). When  $C_i = B$ , it means the opposite (i.e. it is preferred for  $A_i$  to be as large as possible).

To arrange those seats, Mr. Dengklek will fill a bag with a certain amount of tickets and ask each person in the line whether they want to switch their seat ticket with one of the tickets in the bag.

Formally, Mr. Dengklek will do the operation as follows:

- First, Mr. Dengklek will fill bag  $G$  with  $k$  tickets.
- $G_i$  is defined as the row to which the  $i$ -th ticket in the bag is referring.
- Then, Mr. Dengklek will ask the person in the order  $0, 1, \dots, N - 1$ .
- The  $i$ -th person with preference **B** will swap his ticket with the farthest one from the stage if  $A_i < \max(G_0, G_1 \dots G_{k-1})$ .
- Conversely, the  $i$ -th person with preference **S** will swap his ticket with the closest one from the stage if  $A_i > \min(G_0, G_1, \dots, G_{k-1})$ .

Mr. Dengklek has prepared  $Q$  questions for you. In the  $k$ -th question, he starts with a bag that contains  $[B_0, B_1, \dots, B_{k-1}]$  tickets, and wants to know the value of  $\sum_{i=0}^{k-1} G_i$  after the operation.

### Task

You have to implement the following functions:

- `void prepare(int N, vector A, string C, int Q)` — this function is the first function that will be called and it's only called once.
  - $N$ : number of the person in the line.
  - $A$ : an array of length  $N$  describing people's initial tickets.
  - $C$ : a string consisting of  $N$  characters describing people's preferences.
  - $Q$ : number of the question Mr. Dengklek has thought.
- `long long query(const vector &B)` — this function will be called  $Q$  times.
  - $B$ : array of integers describing Mr. Dengklek's question.
  - In the  $i$ -th call,  $B$  will contains  $[B_0, B_1 \dots B_{i-1}]$ .
  - This function should return an integer describing Mr. Dengklek's answer for question  $i$ .

### Constraint

- $1 \leq N, Q \leq 200\,000$ .
- $0 \leq A_i, B_i \leq 10^9$ .
- $|C| = N$ .
- $C_i \in \{\mathbf{S}, \mathbf{B}\}$ .

#### Subtask 1 (10 points)

- $N, Q \leq 1000$ .

#### Subtask 2 (26 points)

- $N, Q \leq 10\,000$ .

#### Subtask 3 (18 points)

- There are no more than 10 indexes of  $i$  with  $C_i = \mathbf{B}$ .

#### Subtask 4 (46 points)

- No additional constraints.

### Sample Interaction

Consider the following case where  $N = 3$ ,  $A = [1, 5, 3]$ ,  $C = \mathbf{SSB}$ ,  $Q = 4$ , and  $B = [5, 3, 2, 9]$ . In this case, the grader will call functions as follows in order:

- `prepare(3, [1, 5, 3], "SSB", 4)`
- `query([5])` — when this function is called it should return 3.
- `query([5, 3])` — when this function is called it should return 8.
- `query([5, 3, 2])` — when this function is called it should return 11.
- `query([5, 3, 2, 9])` — when this function is called it should return 16.

In the second question the iteration proceeds as follows:

- Mr. Dengklek put tickets that refer to seats with row 5 and 3 into bag  $G$ .
- The first person declines Mr. Dengklek's offer.
- The second person swaps his ticket with a ticket from Mr. Dengklek's bag that refers to row 3. Thus,  $G = [5, 5]$  after this iteration.
- The third person swaps his ticket with a ticket from Mr. Dengklek's bag that refers to row 5. Thus,  $G = [3, 5]$  after this iteration.

### Sample Grader

Sample grader will read input in the following format:

```
N
A1 A2 ... AN
C
Q
B1 B2 ... BQ
```

Sample grader will print the output in the following format:

```
ans1 ans2 ... ansQ
```

