

Contest Duration: 2025-08-30(Sat) 22:00 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250830T2100&p1=248>) - 2025-08-30(Sat) 23:40 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250830T2240&p1=248>) (local time) (100 minutes)

[Back to Home \(/home\)](/home)

[🏠 Top \(/contests/abc421\)](/contests/abc421)

[📋 Tasks \(/contests/abc421/tasks\)](/contests/abc421/tasks)

[❓ Clarifications \(/contests/abc421/clarifications\)](/contests/abc421/clarifications)

[📊 Results ▼](#)

[🏆 Standings \(/contests/abc421/standings\)](/contests/abc421/standings)

[🏆 Virtual Standings \(/contests/abc421/standings/virtual\)](/contests/abc421/standings/virtual)

[📖 Editorial \(/contests/abc421/editorial\)](/contests/abc421/editorial)

[💬 Discuss \(https://codeforces.com/blog/entry/145931\)](https://codeforces.com/blog/entry/145931)



## F - Erase between X and Y

[Editorial \(/contests/abc421/tasks/abc421\\_f/editorial\)](/contests/abc421/tasks/abc421_f/editorial)



Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 525 points

### Problem Statement

There is a sequence  $A$ . Initially,  $A = (0)$ . (That is,  $A$  is a sequence of length 1 containing 0 as its only element).

You are given  $Q$  queries to process in order. The  $i$ -th query ( $1 \leq i \leq Q$ ) has one of the following forms:

- 1  $x$ : Insert  $i$  immediately after the location where  $x$  appears in  $A$ . Specifically, let  $A_j$  be the  $j$ -th element of the current  $A$  and  $n$  be the length of  $A$ . For  $p$  such that  $A_p = x$ , update  $A$  to  $(A_1, \dots, A_p, i, A_{p+1}, \dots, A_n)$ . It is guaranteed that  $A$  contains  $x$  immediately before processing this query.
- 2  $x$   $y$ : Remove all elements between  $x$  and  $y$  in  $A$ , and output the sum of the values of the removed elements. Specifically, let  $A_j$  be the  $j$ -th element of the current  $A$  and  $n$  be the length of  $A$ . For  $p$  and  $q$  such that  $A_p = x$  and  $A_q = y$ , output  $A_{\min(p,q)+1} + \dots + A_{\max(p,q)-1}$  and update  $A$  to  $(A_1, \dots, A_{\min(p,q)}, A_{\max(p,q)}, \dots, A_n)$ . It is guaranteed that  $A$  contains both  $x$  and  $y$  immediately before processing this query.

Note that for any sequence of queries, the same value never appears multiple times in  $A$  during the process of handling queries, and thus the position where a value appears in  $A$  is unique (if it exists).

2026-01-02 (Fri)

05:29:30 +11:00

## Constraints

- $1 \leq Q \leq 5 \times 10^5$
- For the  $i$ -th query:
  - If it is a type 1 query:
    - $0 \leq x < i$
    - $A$  contains  $x$  immediately before processing the query.
  - If it is a type 2 query:
    - $0 \leq x < y < i$
    - $A$  contains both  $x$  and  $y$  immediately before processing the query.
- All input values are integers.

## Input

The input is given from Standard Input in the following format:

```
Q
query1
query2
⋮
queryQ
```

Here,  $\text{query}_i$  represents the  $i$ -th query and is given in one of the following forms:

```
1 x
```

```
2 x y
```

## Output

Let  $q$  be the number of type 2 queries. Output  $q$  lines. The  $i$ -th line should contain the value to be output for the  $i$ -th type 2 query.

## Sample Input 1

[Copy](#)[Copy](#)

```
6
1 0
1 1
1 0
2 2 3
1 2
2 0 5
```

## Sample Output 1

[Copy](#)

```
1
5
```

[Copy](#)

Initially,  $A = (0)$ .

- 1st query: Insert 1 immediately after 0.  $A$  becomes  $(0, 1)$ .
- 2nd query: Insert 2 immediately after 1.  $A$  becomes  $(0, 1, 2)$ .
- 3rd query: Insert 3 immediately after 0.  $A$  becomes  $(0, 3, 1, 2)$ .
- 4th query: Remove the elements between 2 and 3, namely 1, and output the sum of the removed values, which is 1.  $A$  becomes  $(0, 3, 2)$ .
- 5th query: Insert 5 immediately after 2.  $A$  becomes  $(0, 3, 2, 5)$ .
- 6th query: Remove the elements between 0 and 5, namely 3, 2, and output the sum of the removed values, which is 5.  $A$  becomes  $(0, 5)$ .

## Sample Input 2

[Copy](#)

```
2
1 0
2 0 1
```

[Copy](#)

## Sample Output 2

[Copy](#)

```
0
```

[Copy](#)

In the 2nd query, we remove all elements between 0 and 1, but there are actually no such elements, so no elements are removed and the output value is 0.

## Sample Input 3

[Copy](#)[Copy](#)

2026-01-02 (Fri)  
05:29:30 +11:00

```
10
1 0
1 1
2 0 2
2 0 2
1 0
1 5
2 0 5
2 2 6
1 6
1 9
```

## Sample Output 3

[Copy](#)

```
1
0
0
0
```

[Copy](#)

[/#telegram](#))

[#url=https%3A%2F%2Fatcoder.jp%2Fcontests%2Fabc421%2Ftasks%2Fabc421\\_f%3Flang%3Den&title=F%20-ind%20Y](#))

---

[Rule \(/contests/abc421/rules\)](/contests/abc421/rules) [Glossary \(/contests/abc421/glossary\)](/contests/abc421/glossary)

[Terms of service \(/tos\)](/tos) [Privacy Policy \(/privacy\)](/privacy) [Information Protection Policy \(/personal\)](/personal) [Company \(/company\)](/company)

[FAQ \(/faq\)](/faq) [Contact \(/contact\)](/contact)

Copyright Since 2012 ©AtCoder Inc. (<http://atcoder.co.jp>) All rights reserved.

2026-01-02 (Fri)  
05:29:30 +11:00