

The Prestige (PRESTIGE)

Robert Angier and his assistant are back in town for another magic trick (nothing as fancy as “The Transported Man”, but it’s still good).

The Setup:

The audience is shown two separate deck of cards. Both have N cards in them, and each deck is arranged in a stack.

Each card of the first deck has the number **1** written on one face, and **-1** on the other face. Initially, the deck is arranged so that all the cards have **1** on upper face, and **-1** on lower.

Each card of the second deck has integers written on both faces. These integers do not have to be the same. The order in which these cards are stacked is given to you.

The Performance:

Robert does one of the following three actions every minute:

- Type 1: He asks some random person in the audience to pick two numbers L and R . Then he picks the cards at positions L to R from the top of the second deck, flips them as a whole, and reinserts them at the same position from where he extracted them. This flipping process has been described below in the 'Flipping' section in more detail. This action is denoted by $1\ L\ R$.
- Type 2: He asks his assistant to pick a number K . Then he picks the first K cards from the top of the first deck, flips them as a whole, and places them back on the same deck. This flipping process has been described below in the 'Flipping' section in more detail. This action is denoted by $2\ K$.
- Type 3: He asks some random person in the audience to pick four numbers A, B, C and D such that $(B - A) == (D - C)$. This action is denoted by $3\ A\ B\ C\ D$. Then he proceeds to perform The Prestige.

The Prestige:

Robert instantly reports the value of $(first[C] * second[A]) + (first[C + 1] * second[A + 1]) + \dots + (first[D] * second[B])$

where $first[i]$ and $second[i]$ are values on the upper faces of i -th card of the first and second decks, respectively.

The Trick:

When asked to pick a number K , the assistant always picks a value at least as large as the one he previously picked (The first time he is asked to pick, he picks some integer ≥ 0).

Flipping:

Suppose you have a deck with 6 cards, with the card at index 1 on the top of the stack, and the card with index 6 at the bottom of the stack. Let the values written on each of the cards be as follows:

Index	Upper Face	Lower Face
1	a	b
2	c	d
3	e	f
4	g	h
5	i	j
6	k	l

After you flip from positions 2 to 5 (ie. doing the operation $1\ 2\ 5$), it will become

Index	Upper Face	Lower Face
1	a	b
2	j	i
3	h	g
4	f	e
5	d	c
6	k	l

In other words, when you perform the action **1 L R** , you 'flip' the cards from L to R , which means that you reverse the order of these cards, and also change the orientation of each of these cards. That is, the old upper face now becomes the new lower face, and vice versa. The action **2 K** simply refers to flipping the first K cards of the first deck.

Robert however has become distracted with Borden's more impressive performances, and hence you have to help him perform this trick. Given the sequence of actions that is to be done, output the answer for every action of the third type.

Input:

- The first line contains N and M , the number of cards in each deck, and the total number of actions Robert does.
- The second line contains N integers, representing the values on the upper face of the second deck, from the top of the deck to bottom.
- The third line contains N integers, representing the values on the lower face of the second deck, from the top of the deck to bottom.
- The following M lines contain descriptions of the actions. Each action can be represented as one of the following (corresponding to the actions described in The Performance section):
 - 1 L R**
 - 2 K**
 - 3 A B C D**

Output:

- For every action of type 3, output its answer in a new line.

Constraints:

- $1 \leq N \leq 200000$
- $1 \leq M \leq 200000$
- $-10^9 \leq \text{integers on the cards} \leq 10^9$
- $1 \leq L \leq R \leq N$
- $0 \leq K \leq N$
- $1 \leq A \leq B \leq N$
- $1 \leq C \leq D \leq N$
- The values of K in the actions of type 2 will be in non-decreasing order.

Sample Input:

```
7 9
3 5 0 2 5 7 4
2 8 4 4 0 1 6
3 1 3 3 5
2 1
1 2 6
1 2 7
2 3
3 1 5 3 7
3 1 7 1 7
1 1 6
3 1 7 1 7
```

Sample Output:

8
16
10
21

Sample Explanation:

- Initially, the first deck is as follows:

Index	Upper Face	Lower Face
1	1	-1
2	1	-1
3	1	-1
4	1	-1
5	1	-1
6	1	-1
7	1	-1

And the second deck is as follows:

Index	Upper Face	Lower Face
1	3	2
2	5	8
3	0	4
4	2	4
5	5	0
6	7	1
7	4	6

- In the first action, you want to find
 $(first[3] * second[1]) + (first[4] * second[2]) + (first[5] * second[3])$
 $= (1 * 3) + (1 * 5) + (1 * 0) = 8$

Hence the first output is 8.

- After the second action, the first deck becomes:

Index	Upper Face	Lower Face
1	-1	1
2	1	-1
3	1	-1
4	1	-1
5	1	-1
6	1	-1
7	1	-1

- After the third action, the second deck becomes:

Index	Upper Face	Lower Face
1	3	2
2	1	7
3	0	5
4	4	2
5	4	0
6	8	5
7	4	6

- After the fourth action, the second deck becomes:

Index	Upper Face	Lower Face
1	3	2
2	6	4
3	5	8
4	0	4
5	2	4
6	5	0
7	7	1

- After the fifth action, the first deck becomes:

Index	Upper Face	Lower Face
1	-1	1
2	-1	1
3	1	-1
4	1	-1
5	1	-1
6	1	-1
7	1	-1

- In the sixth action, you want to find
 $(first[3] * second[1]) + (first[4] * second[2]) + (first[5] * second[3])$
 $+ (first[6] * second[4]) + (first[7] * second[5])$
 $= (1 * 3) + (1 * 6) + (1 * 5) + (1 * 0) + (1 * 2) = 16$
Hence the second output is 16.
- In the seventh action, you want to find
 $(first[1] * second[1]) + (first[2] * second[2]) + (first[3] * second[3])$
 $+ (first[4] * second[4]) + (first[5] * second[5]) + (first[6] * second[6])$
 $+ (first[7] * second[7])$
 $= (-1 * 3) + (-1 * 6) + (1 * 5) + (1 * 0) + (1 * 2) + (1 * 5) + (1 * 7) = 10$
Hence the third output is 10.
- After the eighth action, the second deck becomes:

Index	Upper Face	Lower Face
1	0	5
2	4	2
3	4	0
4	8	5
5	4	6
6	2	3
7	7	1

- In the ninth action, you want to find
 $(first[1] * second[1]) + (first[2] * second[2]) + (first[3] * second[3])$
 $+ (first[4] * second[4]) + (first[5] * second[5]) + (first[6] * second[6])$
 $+ (first[7] * second[7])$
 $= (-1 * 0) + (-1 * 4) + (1 * 4) + (1 * 8) + (1 * 4) + (1 * 2) + (1 * 7) = 21$
Hence the fourth output is 21.