D. Bearish Fanpages

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is a social website with n fanpages, numbered 1 through n. There are also n companies, and the i-th company owns the i-th fanpage.

Recently, the website created a feature called following. Each fanpage must choose exactly one other fanpage to follow.

The website doesn't allow a situation where i follows j and at the same time j follows i. Also, a fanpage can't follow itself.

Let's say that fanpage i follows some other fanpage j_0. Also, let's say that i is followed by k other fanpages $j_1, j_2, \ldots, j_k$. Then, when people visit fanpage i they see ads from $k+2$ distinct companies: $i, j_0, j_1, \ldots, j_k$. Exactly t_i people subscribe (like) the i-th fanpage, and each of them will click exactly one add. For each of $k+1$ companies $j_0, j_1, \ldots, j_k$, exactly people will click their ad. Remaining people will click an ad from company i (the owner of the fanpage).

The total income of the company is equal to the number of people who click ads from this copmany.

Limak and Radewoosh ask you for help. Initially, fanpage i follows fanpage f_i. Your task is to handle q queries of three types:

- `1 i j` — fanpage i follows fanpage j from now. It's guaranteed that i didn't follow j just before the query. Note an extra constraint for the number of queries of this type (below, in the Input section).
- `2 i` — print the total income of the i-th company.
- `3` — print two integers: the smallest income of one company and the biggest income of one company.

Input

The first line of the input contains two integers n and q ($3 \le n \le 100\,000$, $1 \le q \le 100\,000$) — the number of fanpages and the number of queries, respectively.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 10^{12}$) where t_i denotes the number of people subscribing the i-th fanpage.

The third line contains n integers $f_1, f_2, \ldots, f_n$ ($1 \le f_i \le n$). Initially, fanpage i follows fanpage f_i.

Then, q lines follow. The i-th of them describes the i-th query. The first number in the line is an integer $type_i$ ($1 \le type_i \le 3$) — the type of the query.

There will be at most $50\,000$ queries of the first type. There will be at least one query of the second or the third type (so, the output won't be empty).

It's guaranteed that at each moment a fanpage doesn't follow itself, and that no two fanpages follow each other.

Output

For each query of the second type print one integer in a separate line - the total income of the given company. For each query of the third type print two integers in a separate line - the minimum and the maximum total income, respectively.

Example

input
5 12 10 20 30 40 50 2 3 4 5 2 2 1 2 2

```
2 3
2 4
2 5
1 4 2
2 1
2 2
2 3
2 4
2 5
3
```

output

```
10
36
28
40
36
9
57
27
28
29
9 57
```

Note

In the sample test, there are 5 fanpages. The i-th of them has $i \cdot 10$ subscribers.

On drawings, numbers of subscribers are written in circles. An arrow from A to B means that A follows B.

The left drawing shows the initial situation. The first company gets income from its own fanpage, and gets income from the 2-nd fanpage. So, the total income is $5 + 5 = 10$. After the first query ("`2 1`") you should print 10.

The right drawing shows the situation after a query "`1 4 2`" (after which fanpage 4 follows fanpage 2). Then, the first company still gets income 5 from its own fanpage, but now it gets only from the 2-nd fanpage. So, the total income is $5 + 4 = 9$ now.