Codeforces Round #179 (Div. 1)

A. Greg and Array

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Greg has an array $a = a_1, a_2, ..., a_n$ and m operations. Each operation looks as: $l_i, r_i, d_i, (1 \le l_i \le r_i \le n)$. To apply operation i to the array means to increase all array elements with numbers $l_i, l_i + 1, ..., r_i$ by value d_i.

Greg wrote down k queries on a piece of paper. Each query has the following form: $x_i, y_i, (1 \le x_i \le y_i \le m)$. That means that one should apply operations with numbers $x_i, x_i + 1, ..., y_i$ to the array.

Now Greg is wondering, what the array a will be after all the queries are executed. Help Greg.

Input

The first line contains integers n, m, k $(1 \le n, m, k \le 10^5)$. The second line contains n integers: $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^5)$ — the initial array.

Next m lines contain operations, the operation number i is written as three integers: $l_i, r_i, d_i, (1 \le l_i \le r_i \le n), (0 \le d_i \le 10^5)$.

Next k lines contain the queries, the query number i is written as two integers: $x_i, y_i, (1 \le x_i \le y_i \le m)$.

The numbers in the lines are separated by single spaces.

Output

On a single line print n integers $a_1, a_2, ..., a_n$ — the array after executing all the queries. Separate the printed numbers by spaces.

Please, do not use the `%lld` specifier to read or write 64-bit integers in *C++*. It is preferred to use the `cin`, `cout` streams of the `%I64d` specifier.

Sample test(s)

input

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

output

```
9 18 17
```

input

```
1 1 1
1
1 1 1
1 1
```

output

```
2
```

input

```
4 3 6
1 2 3 4
1 2 1
2 3 2
3 4 4
1 2
1 3
2 3
1 2
1 3
2 3
```

output

```
5 18 31 20
```

B. Greg and Graph

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Greg has a weighed directed graph, consisting of n vertices. In this graph any pair of distinct vertices has an edge between them in both directions. Greg loves playing with the graph and now he has invented a new game:

- The game consists of n steps.
- On the i-th step Greg removes vertex number x_i from the graph. As Greg removes a vertex, he also removes all the edges that go in and out of this vertex.
- Before executing each step, Greg wants to know the sum of lengths of the shortest paths between all pairs of the remaining vertices. The shortest path can go through any remaining vertex. In other words, if we assume that $d(i, v, u)$ is the shortest path between vertices v and u in the graph that formed before deleting vertex x_i, then Greg wants to know the value of the following sum: $\sum_{v,u,v\neq u} d(i, v, u)$.

Help Greg, print the value of the required sum before each step.

Input

The first line contains integer n $(1 \le n \le 500)$ — the number of vertices in the graph.

Next n lines contain n integers each — the graph adjacency matrix: the j-th number in the i-th line a_{ij} $(1 \le a_{ij} \le 10^5, a_{ii} = 0)$ represents the weight of the edge that goes from vertex i to vertex j.

The next line contains n distinct integers: $x_1, x_2, ..., x_n$ $(1 \le x_i \le n)$ — the vertices that Greg deletes.

Output

Print n integers — the i-th number equals the required sum before the i-th step.

Please, do not use the `%lld` specifier to read or write 64-bit integers in *C++*. It is preferred to use the `cin`, `cout` streams of the `%I64d` specifier.

Sample test(s)

input
```
1
0
1
```
output
```
0
```

input
```
2
0 5
4 0
1 2
```
output
```
9 0
```

input
```
4
0 3 1 1
6 0 400 1
2 4 0 1
1 1 1 0
4 1 2 3
```
output
```
17 23 404 0
```

C. Greg and Friends

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Greg and his friends were walking in the forest. Overall there were n people walking, including Greg. Soon he found himself in front of a river. The guys immediately decided to get across the river. Luckily, there was a boat by the river bank, just where the guys were standing. We know that the boat can hold people with the total weight of at most k kilograms.

Greg immediately took a piece of paper and listed there the weights of all people in his group (including himself). It turned out that each person weights either 50 or 100 kilograms. Now Greg wants to know what minimum number of times the boat needs to cross the river to transport the whole group to the other bank. The boat needs at least one person to navigate it from one bank to the other. As the boat crosses the river, it can have any non-zero number of passengers as long as their total weight doesn't exceed k.

Also Greg is wondering, how many ways there are to transport everybody to the other side in the minimum number of boat rides. Two ways are considered distinct if during some ride they have distinct sets of people on the boat.

Help Greg with this problem.

Input

The first line contains two integers n, k $(1 \le n \le 50, 1 \le k \le 5000)$ — the number of people, including Greg, and the boat's weight limit. The next line contains n integers — the people's weights. A person's weight is either 50 kilos or 100 kilos.

You can consider Greg and his friends indexed in some way.

Output

In the first line print an integer — the minimum number of rides. If transporting everyone to the other bank is impossible, print an integer `-1`.

In the second line print the remainder after dividing the number of ways to transport the people in the minimum number of rides by number 1000000007 $(10^9 + 7)$. If transporting everyone to the other bank is impossible, print integer 0.

Sample test(s)

input
1 50 50
output
1 1

input
3 100 50 50 100
output
5 2

input
2 50 50 50
output
-1 0

Note

In the first test Greg walks alone and consequently, he needs only one ride across the river.

In the second test you should follow the plan:

1. transport two 50 kg. people;
2. transport one 50 kg. person back;
3. transport one 100 kg. person;
4. transport one 50 kg. person back;
5. transport two 50 kg. people.

That totals to 5 rides. Depending on which person to choose at step 2, we can get two distinct ways.

D. Greg and Caves

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Greg has a pad. The pad's screen is an $n \times m$ rectangle, each cell can be either black or white. We'll consider the pad rows to be numbered with integers from 1 to n from top to bottom. Similarly, the pad's columns are numbered with integers from 1 to m from left to right.

Greg thinks that the pad's screen displays a cave if the following conditions hold:

- There is a segment $[l, r]$ $(1 \le l \le r \le n)$, such that each of the rows $l, l+1, ..., r$ has exactly two black cells and all other rows have only white cells.
- There is a row number t $(l \le t \le r)$, such that for all pairs of rows with numbers i and j $(l \le i \le j \le t)$ the set of columns between the black cells in row i **(with the columns where is these black cells)** is the subset of the set of columns between the black cells in row j **(with the columns where is these black cells)**. Similarly, for all pairs of rows with numbers i and j $(t \le i \le j \le r)$ the set of columns between the black cells in row j **(with the columns where is these black cells)** is the subset of the set of columns between the black cells in row i **(with the columns where is these black cells)**.

Greg wondered, how many ways there are to paint a cave on his pad. Two ways can be considered distinct if there is a cell that has distinct colors on the two pictures.

Help Greg.

Input
The first line contains two integers n, m — the pad's screen size $(1 \le n, m \le 2000)$.

Output
In the single line print the remainder after dividing the answer to the problem by 1000000007 $(10^9 + 7)$.

Sample test(s)

input
1 1
output
0

input
4 4
output
485

input
3 5
output
451

E. Yaroslav and Points

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

Yaroslav has n points that lie on the Ox axis. The coordinate of the first point is x_1, the coordinate of the second point is x_2, ..., the coordinate of the n-th point is $-x_n$. Now Yaroslav wants to execute m queries, each of them is of one of the two following types:

1. Move the p_j-th point from position x_{p_j} to position $x_{p_j} + d_j$. At that, it is guaranteed that after executing such query all coordinates of the points will be distinct.
2. Count the sum of distances between all pairs of points that lie on the segment $[l_j, r_j]$ $(l_j \le r_j)$. In other words, you should count the sum of: $\sum_{l_j \le x_p \le x_q \le r_j} (x_q - x_p)$.

Help Yaroslav.

Input

The first line contains integer n — the number of points $(1 \le n \le 10^5)$. The second line contains distinct integers $x_1, x_2, ..., x_n$ — the coordinates of points $(|x_i| \le 10^9)$.

The third line contains integer m — the number of queries $(1 \le m \le 10^5)$. The next m lines contain the queries. The j-th line first contains integer t_j $(1 \le t_j \le 2)$ — the query type. If $t_j = 1$, then it is followed by two integers p_j and d_j $(1 \le p_j \le n, |d_j| \le 1000)$. If $t_j = 2$, then it is followed by two integers l_j and r_j $(-10^9 \le l_j \le r_j \le 10^9)$.

It is guaranteed that at any moment all the points have distinct coordinates.

Output

For each type 2 query print the answer on a single line. Print the answers in the order, in which the queries follow in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in *C++*. It is preferred to use the `cin`, `cout` streams of the `%I64d` specifier.

Sample test(s)

input

```
8
36 50 28 -75 40 -60 -95 -48
20
2 -61 29
1 5 -53
1 1 429
1 5 130
2 -101 -71
2 -69 53
1 1 404
1 5 518
2 -101 53
2 50 872
1 1 -207
2 -99 -40
1 7 -389
1 6 -171
1 2 464
1 7 -707
1 1 -730
1 1 560
2 635 644
1 7 -677
```

output

```
176
20
406
1046
1638
156
0
```