Codeforces Round #179 (Div. 2)

A. Yaroslav and Permutations

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

Yaroslav has an array that consists of n integers. In one second Yaroslav can swap two neighboring array elements. Now Yaroslav is wondering if he can obtain an array where any two neighboring elements would be distinct in a finite time.

Help Yaroslav.

Input

The first line contains integer n $(1 \le n \le 100)$ — the number of elements in the array. The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 1000)$ — the array elements.

Output

In the single line print "YES" (without the quotes) if Yaroslav can obtain the array he needs, and "NO" (without the quotes) otherwise.

Sample test(s)

input

```
1
1
```

output

```
YES
```

input

```
3
1 1 2
```

output

```
YES
```

input

```
4
7 7 7 7
```

output

```
NO
```

Note

In the first sample the initial array fits well.

In the second sample Yaroslav can get array: $1, 2, 1$. He can swap the last and the second last elements to obtain it.

In the third sample Yarosav can't get the array he needs.

B. Yaroslav and Two Strings

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

Yaroslav thinks that two strings s and w, consisting of digits and having length n are non-comparable if there are two numbers, i and j $(1 \le i, j \le n)$, such that $s_i > w_i$ and $s_j < w_j$. Here sign s_i represents the i-th digit of string s, similarly, w_j represents the j-th digit of string w.

A string's template is a string that consists of digits and question marks ("`?`").

Yaroslav has two string templates, each of them has length n. Yaroslav wants to count the number of ways to replace all question marks by some integers in both templates, so as to make the resulting strings incomparable. Note that the obtained strings can contain leading zeroes and that distinct question marks can be replaced by distinct or the same integers.

Help Yaroslav, calculate the remainder after dividing the described number of ways by 1000000007 $(10^9 + 7)$.

Input

The first line contains integer n $(1 \le n \le 10^5)$ — the length of both templates. The second line contains the first template — a string that consists of digits and characters "`?`". The string's length equals n. The third line contains the second template in the same format.

Output

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

Sample test(s)

input

```
2
90
09
```

output

```
1
```

input

```
2
11
55
```

output

```
0
```

input

```
5
?????
?????
```

output

```
993531194
```

Note

The first test contains no question marks and both strings are incomparable, so the answer is 1.

The second test has no question marks, but the given strings are comparable, so the answer is 0.

C. 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
```

D. 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
```

E. 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.