A. Dawid and Bags of Candies

1 second, 256 megabytes

Dawid has four bags of candies. The i-th of them contains a_i candies. Also, Dawid has two friends. He wants to give each bag to one of his two friends. Is it possible to distribute the bags in such a way that each friend receives the same amount of candies in total?

Note, that you can't keep bags for yourself or throw them away, each bag should be given to one of the friends.

Input

The only line contains four integers a_1, a_2, a_3 and a_4 ($1 \le a_i \le 100$) — the numbers of candies in each bag.

Output

Output YES if it's possible to give the bags to Dawid's friends so that both friends receive the same amount of candies, or NO otherwise. Each character can be printed in any case (either uppercase or lowercase).

input
```
1 7 11 5
```
output
```
YES
```

input
```
7 3 2 5
```
output
```
NO
```

In the first sample test, Dawid can give the first and the third bag to the first friend, and the second and the fourth bag to the second friend. This way, each friend will receive 12 candies.

In the second sample test, it's impossible to distribute the bags.

B. Ania and Minimizing

1 second, 256 megabytes

Ania has a large integer S. Its decimal representation has length n and doesn't contain any leading zeroes. Ania is allowed to change at most k digits of S. She wants to do it in such a way that S still won't contain any leading zeroes and it'll be minimal possible. What integer will Ania finish with?

Input

The first line contains two integers n and k ($1 \le n \le 200\,000$, $0 \le k \le n$) — the number of digits in the decimal representation of S and the maximum allowed number of changed digits.

The second line contains the integer S. It's guaranteed that S has exactly n digits and doesn't contain any leading zeroes.

Output

Output the minimal possible value of S which Ania can end with. Note that the resulting integer should also have n digits.

input
```
5 3
51528
```
output
```
10028
```

input
```
3 2
102
```
output
```
100
```

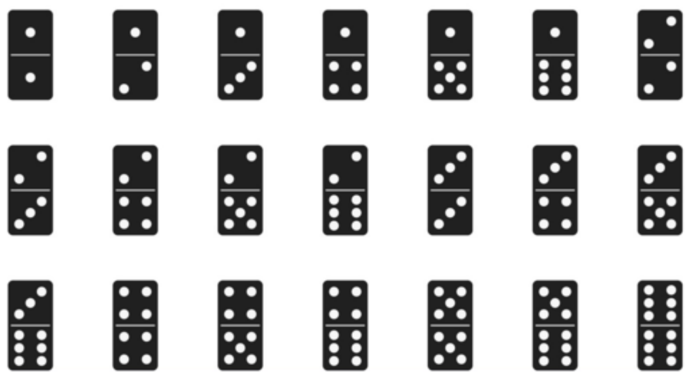
input
```
1 1
1
```
output
```
0
```

A number has leading zeroes if it consists of at least two digits and its first digit is 0. For example, numbers 00, 00069 and 0101 have leading zeroes, while 0, 3000 and 1010 don't have leading zeroes.

C. Anadi and Domino

2 seconds, 256 megabytes

Anadi has a set of dominoes. Every domino has two parts, and each part contains some dots. For every a and b such that $1 \leq a \leq b \leq 6$, there is exactly one domino with a dots on one half and b dots on the other half. The set contains exactly 21 dominoes. Here is an exact illustration of his set:



Also, Anadi has an undirected graph without self-loops and multiple edges. He wants to choose some dominoes and place them on the edges of this graph. He can use at most one domino of each type. Each edge can fit at most one domino. It's not necessary to place a domino on each edge of the graph.

When placing a domino on an edge, he also chooses its direction. In other words, one half of any placed domino must be directed toward one of the endpoints of the edge and the other half must be directed toward the other endpoint. There's a catch: if there are multiple halves of dominoes directed toward the same vertex, each of these halves must contain the same number of dots.

How many dominoes at most can Anadi place on the edges of his graph?

Input

The first line contains two integers n and m ($1 \leq n \leq 7$, $0 \leq m \leq \frac{n\cdot(n-1)}{2}$) — the number of vertices and the number of edges in the graph.

The next m lines contain two integers each. Integers in the i-th line are a_i and b_i ($1 \leq a, b \leq n$, $a \neq b$) and denote that there is an edge which connects vertices a_i and b_i.

The graph might be disconnected. It's however guaranteed that the graph doesn't contain any self-loops, and that there is at most one edge between any pair of vertices.

Output

Output one integer which denotes the maximum number of dominoes which Anadi can place on the edges of the graph.

input

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

output

```
4
```

input

```
7 0
```

output

```
0
```

input

```
3 1
1 3
```

output

```
1
```

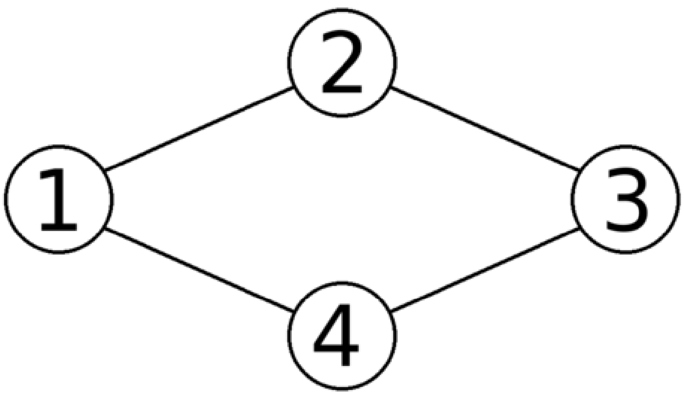
input

```
7 21
1 2
1 3
1 4
1 5
1 6
1 7
2 3
2 4
2 5
2 6
2 7
3 4
3 5
3 6
3 7
4 5
4 6
4 7
5 6
5 7
6 7
```

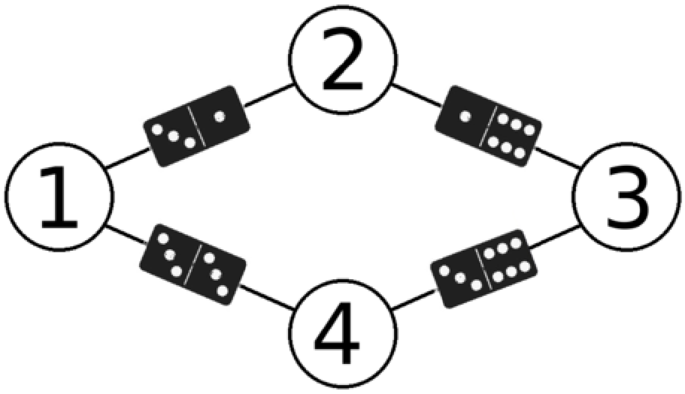
output

```
16
```

Here is an illustration of Anadi's graph from the first sample test:



And here is one of the ways to place a domino on each of its edges:



Note that each vertex is faced by the halves of dominoes with the same number of dots. For instance, all halves directed toward vertex 1 have three dots.

D. Marcin and Training Camp

3 seconds, 256 megabytes

Marcin is a coach in his university. There are n students who want to attend a training camp. Marcin is a smart coach, so he wants to send only the students that can work calmly with each other.

Let's focus on the students. They are indexed with integers from 1 to n. Each of them can be described with two integers a_i and b_i; b_i is equal to the skill level of the i-th student (the higher, the better). Also, there are 60 known algorithms, which are numbered with integers from 0 to 59. If the i-th student knows the j-th algorithm, then the j-th bit (2^j) is set in the binary representation of a_i. Otherwise, this bit is not set.

Student x thinks that he is better than student y if and only if x knows some algorithm which y doesn't know. Note that two students can think that they are better than each other. A group of students can work together calmly if no student in this group thinks that he is better than everyone else in this group.

Marcin wants to send a group of at least two students which will work together calmly and will have the maximum possible sum of the skill levels. What is this sum?

Input

The first line contains one integer n ($1 \le n \le 7000$) — the number of students interested in the camp.

The second line contains n integers. The i-th of them is a_i ($0 \le a_i < 2^{60}$).

The third line contains n integers. The i-th of them is b_i ($1 \le b_i \le 10^9$).

Output

Output one integer which denotes the maximum sum of b_i over the students in a group of students which can work together calmly. If no group of at least two students can work together calmly, print `0`.

input

```
4
3 2 3 6
2 8 5 10
```

output

```
15
```

input

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

output

```
0
```

input

```
1
0
1
```

output

```
0
```

In the first sample test, it's optimal to send the first, the second and the third student to the camp. It's also possible to send only the first and the third student, but they'd have a lower sum of b_i.

In the second test, in each group of at least two students someone will always think that he is better than everyone else in the subset.

E. Kamil and Making a Stream

4 seconds, 768 megabytes

Kamil likes streaming the competitive programming videos. His MeTube channel has recently reached 100 million subscribers. In order to celebrate this, he posted a video with an interesting problem he couldn't solve yet. Can you help him?

You're given a tree — a connected undirected graph consisting of n vertices connected by $n-1$ edges. The tree is rooted at vertex 1. A vertex u is called an *ancestor* of v if it lies on the shortest path between the root and v. In particular, a vertex is an ancestor of itself.

Each vertex v is assigned its *beauty* x_v — a non-negative integer not larger than 10^{12}. This allows us to define the beauty of a path. Let u be an ancestor of v. Then we define the beauty $f(u, v)$ as the greatest common divisor of the beauties of all vertices on the shortest path between u and v. Formally, if $u = t_1, t_2, t_3, \dots, t_k = v$ are the vertices on the shortest path between u and v, then $f(u, v) = \gcd(x_{t_1}, x_{t_2}, \dots, x_{t_k})$. Here, $\gcd$ denotes the greatest common divisor of a set of numbers. In particular, $f(u, u) = \gcd(x_u) = x_u$.

Your task is to find the sum

$$\sum_{u \text{ is an ancestor of } v} f(u, v).$$

As the result might be too large, please output it modulo $10^9 + 7$.

Note that for each y, $\gcd(0, y) = \gcd(y, 0) = y$. In particular, $\gcd(0, 0) = 0$.

Input

The first line contains a single integer n $(2 \le n \le 100\,000)$ — the number of vertices in the tree.

The following line contains n integers $x_1, x_2, \dots, x_n$ $(0 \le x_i \le 10^{12})$. The value x_v denotes the beauty of vertex v.

The following $n-1$ lines describe the edges of the tree. Each of them contains two integers a, b $(1 \le a, b \le n, a \ne b)$ — the vertices connected by a single edge.

Output

Output the sum of the beauties on all paths (u, v) such that u is ancestor of v. This sum should be printed modulo $10^9 + 7$.

input

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

output

```
42
```

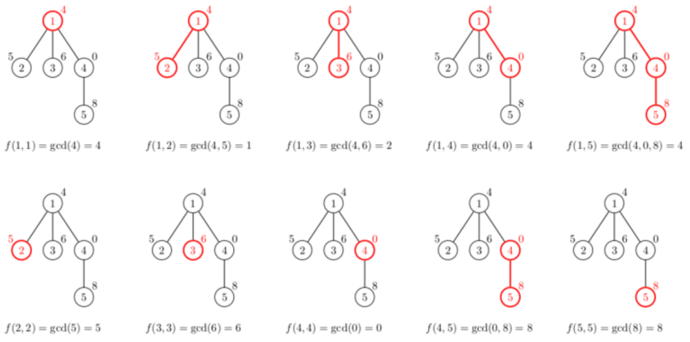
input

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

output

```
30
```

The following figure shows all 10 possible paths for which one endpoint is an ancestor of another endpoint. The sum of beauties of all these paths is equal to 42:



F. Konrad and Company Evaluation

4 seconds, 256 megabytes

Konrad is a Human Relations consultant working for VoltModder, a large electrical equipment producer. Today, he has been tasked with evaluating the level of happiness in the company.

There are n people working for VoltModder, numbered from 1 to n. Each employee earns a different amount of money in the company — initially, the i-th person earns i rubles per day.

On each of q following days, the salaries will be revised. At the end of the i-th day, employee v_i will start earning $n+i$ rubles per day and will become the best-paid person in the company. The employee will keep his new salary until it gets revised again.

Some pairs of people don't like each other. This creates a great psychological danger in the company. Formally, if two people a and b dislike each other and a earns more money than b, employee a will brag about this to b. A *dangerous triple* is a triple of three employees a, b and c, such that a brags to b, who in turn brags to c. If a dislikes b, then b dislikes a.

At the beginning of each day, Konrad needs to evaluate the number of *dangerous triples* in the company. Can you help him do it?

Input

The first line contains two integers n and m $(1 \le n \le 100\,000$, $0 \le m \le 100\,000)$ — the number of employees in the company and the number of pairs of people who don't like each other. Each of the following m lines contains two integers a_i, b_i $(1 \le a_i, b_i \le n$, $a_i \ne b_i)$ denoting that employees a_i and b_i hate each other (that is, a_i dislikes b_i and b_i dislikes a_i). Each such relationship will be mentioned exactly once.

The next line contains an integer q $(0 \le q \le 100\,000)$ — the number of salary revisions. The i-th of the following q lines contains a single integer v_i $(1 \le v_i \le n)$ denoting that at the end of the i-th day, employee v_i will earn the most.

Output

Output $q+1$ integers. The i-th of them should contain the number of *dangerous triples* in the company at the beginning of the i-th day.

input

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

output

```
4
3
2
```

input

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

output

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

Consider the first sample test. The i-th row in the following image shows the structure of the company at the beginning of the i-th day. A directed edge from a to b denotes that employee a brags to employee b. The dangerous triples are marked by highlighted edges.