A1. Special Task

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Special Agent Smart Beaver works in a secret research department of ABBYY. He's been working there for a long time and is satisfied with his job, as it allows him to eat out in the best restaurants and order the most expensive and exotic wood types there.

The content special agent has got an important task: to get the latest research by British scientists on the English Language. These developments are encoded and stored in a large safe. The Beaver's teeth are strong enough, so the authorities assured that upon arriving at the place the beaver won't have any problems with opening the safe.

And he finishes his aspen sprig and leaves for this important task. Of course, the Beaver arrived at the location without any problems, but alas. He can't open the safe with his strong and big teeth. At this point, the Smart Beaver get a call from the headquarters and learns that opening the safe with the teeth is not necessary, as a reliable source has sent the following information: the safe code consists of digits and has no leading zeroes. There also is a special hint, which can be used to open the safe. The hint is string s with the following structure:

- if s_i = "?", then the digit that goes i-th in the safe code can be anything (between 0 to 9, inclusively);
- if s_i is a digit (between 0 to 9, inclusively), then it means that there is digit s_i on position i in code;
- if the string contains letters from "A" to "J", then all positions with the same letters must contain the same digits and the positions with distinct letters must contain distinct digits.
- The length of the safe code coincides with the length of the hint.

For example, hint "?JGJ9" has such matching safe code variants: "51919", "55959", "12329", "93539" and so on, and has wrong variants such as: "56669", "00111", "03539" and "13666".

After receiving such information, the authorities change the plan and ask the special agents to work quietly and gently and not to try to open the safe by mechanical means, and try to find the password using the given hint.

At a special agent school the Smart Beaver was the fastest in his platoon finding codes for such safes, but now he is not in that shape: the years take their toll ... Help him to determine the number of possible variants of the code to the safe, matching the given hint. After receiving this information, and knowing his own speed of entering codes, the Smart Beaver will be able to determine whether he will have time for tonight's show "Beavers are on the trail" on his favorite TV channel, or he should work for a sleepless night...

Input

The first line contains string s — the hint to the safe code. String s consists of the following characters: `?, 0-9, A-J`. It is guaranteed that the first character of string s doesn't equal to character 0.

The input limits for scoring 30 points are (subproblem A1):

- $1 \le |s| \le 5$.

The input limits for scoring 100 points are (subproblems A1+A2):

- $1 \le |s| \le 10^5$.

Here $|s|$ means the length of string s.

Output

Print the number of codes that match the given hint.

Examples

input
AJ
output
81

input
1?AA

output
100

A2. Special Task

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Special Agent Smart Beaver works in a secret research department of ABBYY. He's been working there for a long time and is satisfied with his job, as it allows him to eat out in the best restaurants and order the most expensive and exotic wood types there.

The content special agent has got an important task: to get the latest research by British scientists on the English Language. These developments are encoded and stored in a large safe. The Beaver's teeth are strong enough, so the authorities assured that upon arriving at the place the beaver won't have any problems with opening the safe.

And he finishes his aspen sprig and leaves for this important task. Of course, the Beaver arrived at the location without any problems, but alas. He can't open the safe with his strong and big teeth. At this point, the Smart Beaver get a call from the headquarters and learns that opening the safe with the teeth is not necessary, as a reliable source has sent the following information: the safe code consists of digits and has no leading zeroes. There also is a special hint, which can be used to open the safe. The hint is string s with the following structure:

- if s_i = "?", then the digit that goes i-th in the safe code can be anything (between 0 to 9, inclusively);
- if s_i is a digit (between 0 to 9, inclusively), then it means that there is digit s_i on position i in code;
- if the string contains letters from "A" to "J", then all positions with the same letters must contain the same digits and the positions with distinct letters must contain distinct digits.
- The length of the safe code coincides with the length of the hint.

For example, hint "?JGJ9" has such matching safe code variants: "51919", "55959", "12329", "93539" and so on, and has wrong variants such as: "56669", "00111", "03539" and "13666".

After receiving such information, the authorities change the plan and ask the special agents to work quietly and gently and not to try to open the safe by mechanical means, and try to find the password using the given hint.

At a special agent school the Smart Beaver was the fastest in his platoon finding codes for such safes, but now he is not in that shape: the years take their toll ... Help him to determine the number of possible variants of the code to the safe, matching the given hint. After receiving this information, and knowing his own speed of entering codes, the Smart Beaver will be able to determine whether he will have time for tonight's show "Beavers are on the trail" on his favorite TV channel, or he should work for a sleepless night...

Input

The first line contains string s — the hint to the safe code. String s consists of the following characters: `?, 0-9, A-J`. It is guaranteed that the first character of string s doesn't equal to character 0.

The input limits for scoring 30 points are (subproblem A1):

- $1 \le |s| \le 5$.

The input limits for scoring 100 points are (subproblems A1+A2):

- $1 \le |s| \le 10^5$.

Here $|s|$ means the length of string s.

Output

Print the number of codes that match the given hint.

Examples

input
AJ
output
81

input
1?AA
output
100

B1. EKG

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the rush of modern life, people often forget how beautiful the world is. The time to enjoy those around them is so little that some even stand in queues to several rooms at the same time in the clinic, running from one queue to another.

(Cultural note: standing in huge and disorganized queues for hours is a native tradition in Russia, dating back to the Soviet period. Queues can resemble crowds rather than lines. Not to get lost in such a queue, a person should follow a strict survival technique: you approach the queue and ask who the last person is, somebody answers and you join the crowd. Now you're the last person in the queue till somebody else shows up. You keep an eye on the one who was last before you as he is your only chance to get to your destination) I'm sure many people have had the problem when a stranger asks who the last person in the queue is and even dares to hint that he will be the last in the queue and then bolts away to some unknown destination. These are the representatives of the modern world, in which the ratio of lack of time is so great that they do not even watch foreign top-rated TV series. Such people often create problems in queues, because the newcomer does not see the last person in the queue and takes a place after the "virtual" link in this chain, wondering where this legendary figure has left.

The Smart Beaver has been ill and he's made an appointment with a therapist. The doctor told the Beaver the sad news in a nutshell: it is necessary to do an electrocardiogram. The next day the Smart Beaver got up early, put on the famous TV series on download (three hours till the download's complete), clenched his teeth and bravely went to join a queue to the electrocardiogram room, which is notorious for the biggest queues at the clinic.

Having stood for about three hours in the queue, the Smart Beaver realized that many beavers had not seen who was supposed to stand in the queue before them and there was a huge mess. He came up to each beaver in the ECG room queue and asked who should be in front of him in the queue. If the beaver did not know his correct position in the queue, then it might be his turn to go get an ECG, or maybe he should wait for a long, long time...

As you've guessed, the Smart Beaver was in a hurry home, so he gave you all the necessary information for you to help him to determine what his number in the queue can be.

Input

The first line contains two integers n ($1 \le n \le 10^3$) and x ($1 \le x \le n$) — the number of beavers that stand in the queue and the Smart Beaver's number, correspondingly. All willing to get to the doctor are numbered from 1 to n.

The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le n$) — the number of the beaver followed by the i-th beaver. If $a_i = 0$, then the i-th beaver doesn't know who is should be in front of him. It is guaranteed that values a_i are correct. That is there is no cycles in the dependencies. And any beaver is followed by at most one beaver in the queue.

The input limits for scoring 30 points are (subproblem B1):

- It is guaranteed that the number of zero elements a_i doesn't exceed 20.

The input limits for scoring 100 points are (subproblems B1+B2):

- The number of zero elements a_i is arbitrary.

Output

Print all possible positions of the Smart Beaver in the line in the increasing order.

Examples

input
6 1 2 0 4 0 6 0
output
2 4 6

input
6 2 2 3 0 5 6 0
output
2 5

input
4 1 0 0 0 0

output

```
1
2
3
4
```

input

```
6 2
0 0 1 0 4 5
```

output

```
1
3
4
6
```

Note

Picture for the fourth test.

B2. EKG

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the rush of modern life, people often forget how beautiful the world is. The time to enjoy those around them is so little that some even stand in queues to several rooms at the same time in the clinic, running from one queue to another.

(Cultural note: standing in huge and disorganized queues for hours is a native tradition in Russia, dating back to the Soviet period. Queues can resemble crowds rather than lines. Not to get lost in such a queue, a person should follow a strict survival technique: you approach the queue and ask who the last person is, somebody answers and you join the crowd. Now you're the last person in the queue till somebody else shows up. You keep an eye on the one who was last before you as he is your only chance to get to your destination) I'm sure many people have had the problem when a stranger asks who the last person in the queue is and even dares to hint that he will be the last in the queue and then bolts away to some unknown destination. These are the representatives of the modern world, in which the ratio of lack of time is so great that they do not even watch foreign top-rated TV series. Such people often create problems in queues, because the newcomer does not see the last person in the queue and takes a place after the "virtual" link in this chain, wondering where this legendary figure has left.

The Smart Beaver has been ill and he's made an appointment with a therapist. The doctor told the Beaver the sad news in a nutshell: it is necessary to do an electrocardiogram. The next day the Smart Beaver got up early, put on the famous TV series on download (three hours till the download's complete), clenched his teeth and bravely went to join a queue to the electrocardiogram room, which is notorious for the biggest queues at the clinic.

Having stood for about three hours in the queue, the Smart Beaver realized that many beavers had not seen who was supposed to stand in the queue before them and there was a huge mess. He came up to each beaver in the ECG room queue and asked who should be in front of him in the queue. If the beaver did not know his correct position in the queue, then it might be his turn to go get an ECG, or maybe he should wait for a long, long time...

As you've guessed, the Smart Beaver was in a hurry home, so he gave you all the necessary information for you to help him to determine what his number in the queue can be.

Input

The first line contains two integers n ($1 \le n \le 10^3$) and x ($1 \le x \le n$) — the number of beavers that stand in the queue and the Smart Beaver's number, correspondingly. All willing to get to the doctor are numbered from 1 to n.

The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le n$) — the number of the beaver followed by the i-th beaver. If $a_i = 0$, then the i-th beaver doesn't know who is should be in front of him. It is guaranteed that values a_i are correct. That is there is no cycles in the dependencies. And any beaver is followed by at most one beaver in the queue.

The input limits for scoring 30 points are (subproblem B1):

- It is guaranteed that the number of zero elements a_i doesn't exceed 20.

The input limits for scoring 100 points are (subproblems B1+B2):

- The number of zero elements a_i is arbitrary.

Output

Print all possible positions of the Smart Beaver in the line in the increasing order.

Examples

input
6 1 2 0 4 0 6 0
output
2 4 6

input
6 2 2 3 0 5 6 0
output
2 5

input
4 1 0 0 0 0

output

```
1
2
3
4
```

input

```
6 2
0 0 1 0 4 5
```

output

```
1
3
4
6
```

Note

Picture for the fourth test.

C1. Tidying Up

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Smart Beaver is careful about his appearance and pays special attention to shoes so he has a huge number of pairs of shoes from the most famous brands of the forest. He's trying to handle his shoes carefully so that each pair stood side by side. But by the end of the week because of his very active lifestyle in his dressing room becomes a mess.

Smart Beaver from ABBYY is not only the brightest beaver in the area, but he also is the most domestically oriented. For example, on Mondays the Smart Beaver cleans everything in his home.

It's Monday morning. Smart Beaver does not want to spend the whole day cleaning, besides, there is much in to do and it's the gym day, so he wants to clean up as soon as possible. Now the floors are washed, the dust is wiped off — it's time to clean up in the dressing room. But as soon as the Smart Beaver entered the dressing room, all plans for the day were suddenly destroyed: chaos reigned there and it seemed impossible to handle, even in a week. Give our hero some hope: tell him what is the minimum number of shoes need to change the position to make the dressing room neat.

The dressing room is rectangular and is divided into $n \times m$ equal squares, each square contains exactly one shoe. Each pair of shoes has a unique number that is integer from 1 to $\frac{n \cdot m}{2}$, more formally, a square with coordinates (i, j) contains an integer number of the pair which is lying on it. The Smart Beaver believes that the dressing room is neat only when each pair of sneakers lies together. We assume that the pair of sneakers in squares (i_1, j_1) and (i_2, j_2) lies together if $|i_1 - i_2| + |j_1 - j_2| = 1$.

Input

The first line contains two space-separated integers n and m. They correspond to the dressing room size. Next n lines contain m space-separated integers each. Those numbers describe the dressing room. Each number corresponds to a snicker.

It is guaranteed that:

- $n \cdot m$ is even.
- All numbers, corresponding to the numbers of pairs of shoes in the dressing room, will lie between 1 and $\frac{n \cdot m}{2}$.
- Each number from 1 to $\frac{n \cdot m}{2}$ will occur exactly twice.

The input limits for scoring 30 points are (subproblem C1):

- $2 \le n, m \le 8$.

The input limits for scoring 100 points are (subproblems C1+C2):

- $2 \le n, m \le 80$.

Output

Print exactly one integer — the minimum number of the sneakers that need to change their location.

Examples

input
2 3 1 1 2 2 3 3
output
2

input
3 4 1 3 2 6 2 1 5 6 4 4 5 3
output
4

Note

The second sample.

C2. Tidying Up

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Smart Beaver is careful about his appearance and pays special attention to shoes so he has a huge number of pairs of shoes from the most famous brands of the forest. He's trying to handle his shoes carefully so that each pair stood side by side. But by the end of the week because of his very active lifestyle in his dressing room becomes a mess.

Smart Beaver from ABBYY is not only the brightest beaver in the area, but he also is the most domestically oriented. For example, on Mondays the Smart Beaver cleans everything in his home.

It's Monday morning. Smart Beaver does not want to spend the whole day cleaning, besides, there is much in to do and it's the gym day, so he wants to clean up as soon as possible. Now the floors are washed, the dust is wiped off — it's time to clean up in the dressing room. But as soon as the Smart Beaver entered the dressing room, all plans for the day were suddenly destroyed: chaos reigned there and it seemed impossible to handle, even in a week. Give our hero some hope: tell him what is the minimum number of shoes need to change the position to make the dressing room neat.

The dressing room is rectangular and is divided into $n \times m$ equal squares, each square contains exactly one shoe. Each pair of shoes has a unique number that is integer from 1 to $\frac{n \cdot m}{2}$, more formally, a square with coordinates (i, j) contains an integer number of the pair which is lying on it. The Smart Beaver believes that the dressing room is neat only when each pair of sneakers lies together. We assume that the pair of sneakers in squares (i_1, j_1) and (i_2, j_2) lies together if $|i_1 - i_2| + |j_1 - j_2| = 1$.

Input

The first line contains two space-separated integers n and m. They correspond to the dressing room size. Next n lines contain m space-separated integers each. Those numbers describe the dressing room. Each number corresponds to a snicker.

It is guaranteed that:

- $n \cdot m$ is even.
- All numbers, corresponding to the numbers of pairs of shoes in the dressing room, will lie between 1 and $\frac{n \cdot m}{2}$.
- Each number from 1 to $\frac{n \cdot m}{2}$ will occur exactly twice.

The input limits for scoring 30 points are (subproblem C1):

- $2 \le n, m \le 8$.

The input limits for scoring 100 points are (subproblems C1+C2):

- $2 \le n, m \le 80$.

Output

Print exactly one integer — the minimum number of the sneakers that need to change their location.

Examples

input
2 3 1 1 2 2 3 3
output
2

input
3 4 1 3 2 6 2 1 5 6 4 4 5 3
output
4

Note

The second sample.

D1. PE Lesson

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

Smart Beaver decided to be not only smart, but also a healthy beaver! And so he began to attend physical education classes at school X. In this school, physical education has a very creative teacher. One of his favorite warm-up exercises is throwing balls. Students line up. Each one gets a single ball in the beginning. The balls are numbered from 1 to n (by the demand of the inventory commission).

Figure 1. The initial position for $n = 5$.

After receiving the balls the students perform the warm-up exercise. The exercise takes place in a few throws. For each throw the teacher chooses any two arbitrary different students who will participate in it. The selected students throw their balls to each other. Thus, after each throw the students remain in their positions, and the two balls are swapped.

Figure 2. The example of a throw.

In this case there was a throw between the students, who were holding the 2-nd and the 4-th balls. Since the warm-up has many exercises, each of them can only continue for little time. Therefore, for each student we know the maximum number of throws he can participate in. For this lessons maximum number of throws will be 1 or 2.

Note that after all phases of the considered exercise any ball can end up with any student. Smart Beaver decided to formalize it and introduced the concept of the "ball order". The ball order is a sequence of n numbers that correspond to the order of balls in the line. The first number will match the number of the ball of the first from the left student in the line, the second number will match the ball of the second student, and so on. For example, in figure 2 the order of the balls was (1, 2, 3, 4, 5), and after the throw it was (1, 4, 3, 2, 5). Smart beaver knows the number of students and for each student he knows the maximum number of throws in which he can participate. And now he is wondering: what is the number of distinct ways of ball orders by the end of the exercise.

Input

The first line contains a single number n — the number of students in the line and the number of balls. The next line contains exactly n space-separated integers. Each number corresponds to a student in the line (the i-th number corresponds to the i-th from the left student in the line) and shows the number of throws he can participate in.

The input limits for scoring 30 points are (subproblem D1):

- $1 \le n \le 10$.

The input limits for scoring 70 points are (subproblems D1+D2):

- $1 \le n \le 500$.

The input limits for scoring 100 points are (subproblems D1+D2+D3):

- $1 \le n \le 1000000$.

Output

The output should contain a single integer — the number of variants of ball orders after the warm up exercise is complete. As the number can be rather large, print it modulo 1000000007 ($10^9 + 7$).

Examples

input
5 1 2 2 1 2
output
120

input
8 1 2 2 1 2 1 1 2
output
16800

D2. PE Lesson

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

Smart Beaver decided to be not only smart, but also a healthy beaver! And so he began to attend physical education classes at school X. In this school, physical education has a very creative teacher. One of his favorite warm-up exercises is throwing balls. Students line up. Each one gets a single ball in the beginning. The balls are numbered from 1 to n (by the demand of the inventory commission).

Figure 1. The initial position for $n = 5$.

After receiving the balls the students perform the warm-up exercise. The exercise takes place in a few throws. For each throw the teacher chooses any two arbitrary different students who will participate in it. The selected students throw their balls to each other. Thus, after each throw the students remain in their positions, and the two balls are swapped.

Figure 2. The example of a throw.

In this case there was a throw between the students, who were holding the 2-nd and the 4-th balls. Since the warm-up has many exercises, each of them can only continue for little time. Therefore, for each student we know the maximum number of throws he can participate in. For this lessons maximum number of throws will be 1 or 2.

Note that after all phases of the considered exercise any ball can end up with any student. Smart Beaver decided to formalize it and introduced the concept of the "ball order". The ball order is a sequence of n numbers that correspond to the order of balls in the line. The first number will match the number of the ball of the first from the left student in the line, the second number will match the ball of the second student, and so on. For example, in figure 2 the order of the balls was (1, 2, 3, 4, 5), and after the throw it was (1, 4, 3, 2, 5). Smart beaver knows the number of students and for each student he knows the maximum number of throws in which he can participate. And now he is wondering: what is the number of distinct ways of ball orders by the end of the exercise.

Input

The first line contains a single number n — the number of students in the line and the number of balls. The next line contains exactly n space-separated integers. Each number corresponds to a student in the line (the i-th number corresponds to the i-th from the left student in the line) and shows the number of throws he can participate in.

The input limits for scoring 30 points are (subproblem D1):

- $1 \le n \le 10$.

The input limits for scoring 70 points are (subproblems D1+D2):

- $1 \le n \le 500$.

The input limits for scoring 100 points are (subproblems D1+D2+D3):

- $1 \le n \le 1000000$.

Output

The output should contain a single integer — the number of variants of ball orders after the warm up exercise is complete. As the number can be rather large, print it modulo 1000000007 ($10^9 + 7$).

Examples

input
5 1 2 2 1 2
output
120

input
8 1 2 2 1 2 1 1 2
output
16800

D3. PE Lesson

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

Smart Beaver decided to be not only smart, but also a healthy beaver! And so he began to attend physical education classes at school X. In this school, physical education has a very creative teacher. One of his favorite warm-up exercises is throwing balls. Students line up. Each one gets a single ball in the beginning. The balls are numbered from 1 to n (by the demand of the inventory commission).

Figure 1. The initial position for $n = 5$.

After receiving the balls the students perform the warm-up exercise. The exercise takes place in a few throws. For each throw the teacher chooses any two arbitrary different students who will participate in it. The selected students throw their balls to each other. Thus, after each throw the students remain in their positions, and the two balls are swapped.

Figure 2. The example of a throw.

In this case there was a throw between the students, who were holding the 2-nd and the 4-th balls. Since the warm-up has many exercises, each of them can only continue for little time. Therefore, for each student we know the maximum number of throws he can participate in. For this lessons maximum number of throws will be 1 or 2.

Note that after all phases of the considered exercise any ball can end up with any student. Smart Beaver decided to formalize it and introduced the concept of the "ball order". The ball order is a sequence of n numbers that correspond to the order of balls in the line. The first number will match the number of the ball of the first from the left student in the line, the second number will match the ball of the second student, and so on. For example, in figure 2 the order of the balls was (1, 2, 3, 4, 5), and after the throw it was (1, 4, 3, 2, 5). Smart beaver knows the number of students and for each student he knows the maximum number of throws in which he can participate. And now he is wondering: what is the number of distinct ways of ball orders by the end of the exercise.

Input

The first line contains a single number n — the number of students in the line and the number of balls. The next line contains exactly n space-separated integers. Each number corresponds to a student in the line (the i-th number corresponds to the i-th from the left student in the line) and shows the number of throws he can participate in.

The input limits for scoring 30 points are (subproblem D1):

- $1 \le n \le 10$.

The input limits for scoring 70 points are (subproblems D1+D2):

- $1 \le n \le 500$.

The input limits for scoring 100 points are (subproblems D1+D2+D3):

- $1 \le n \le 1000000$.

Output

The output should contain a single integer — the number of variants of ball orders after the warm up exercise is complete. As the number can be rather large, print it modulo 1000000007 ($10^9 + 7$).

Examples

input
5 1 2 2 1 2
output
120

input
8 1 2 2 1 2 1 1 2
output
16800

E1. Summer Homework

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

By the age of three Smart Beaver mastered all arithmetic operations and got this summer homework from the amazed teacher:

You are given a sequence of integers $a_1, a_2, ..., a_n$. Your task is to perform on it m consecutive operations of the following type:

1. For given numbers x_i and v_i assign value v_i to element a_{x_i}.
2. For given numbers l_i and r_i you've got to calculate sum $\sum_{x=l_i}^{r_i} a_x$, where $f_0 = f_1 = 1$ and at $i \geq 2$: $f_i = f_{i-1} + f_{i-2}$.
3. For a group of three numbers $l_i\ r_i\ d_i$ you should increase value a_x by d_i for all x ($l_i \leq x \leq r_i$).

Smart Beaver planned a tour around great Canadian lakes, so he asked you to help him solve the given problem.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$) — the number of integers in the sequence and the number of operations, correspondingly. The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^5$). Then follow m lines, each describes an operation. Each line starts with an integer t_i ($1 \leq t_i \leq 3$) — the operation type:

- if $t_i = 1$, then next follow two integers $x_i\ v_i$ ($1 \leq x_i \leq n, 0 \leq v_i \leq 10^5$);
- if $t_i = 2$, then next follow two integers $l_i\ r_i$ ($1 \leq l_i \leq r_i \leq n$);
- if $t_i = 3$, then next follow three integers $l_i\ r_i\ d_i$ ($1 \leq l_i \leq r_i \leq n, 0 \leq d_i \leq 10^5$).

The input limits for scoring 30 points are (subproblem E1):

- It is guaranteed that n does not exceed 100, m does not exceed 10000 and there will be no queries of the 3-rd type.

The input limits for scoring 70 points are (subproblems E1+E2):

- It is guaranteed that there will be queries of the 1-st and 2-nd type only.

The input limits for scoring 100 points are (subproblems E1+E2+E3):

- No extra limitations.

Output

For each query print the calculated sum modulo 1000000000 (10^9).

Examples

input
5 5 1 3 1 2 4 2 1 4 2 1 5 2 2 4 1 3 10 2 1 5
output
12 32 8 50

input
5 4 1 3 1 2 4 3 1 4 1 2 2 4 1 2 10 2 1 5
output
12 45

E2. Summer Homework

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

By the age of three Smart Beaver mastered all arithmetic operations and got this summer homework from the amazed teacher:

You are given a sequence of integers $a_1, a_2, ..., a_n$. Your task is to perform on it m consecutive operations of the following type:

1. For given numbers x_i and v_i assign value v_i to element a_{x_i}.
2. For given numbers l_i and r_i you've got to calculate sum $\sum_{x=l_i}^{r_i} a_x$, where $f_0 = f_1 = 1$ and at $i \geq 2$: $f_i = f_{i-1} + f_{i-2}$.
3. For a group of three numbers $l_i\ r_i\ d_i$ you should increase value a_x by d_i for all x ($l_i \leq x \leq r_i$).

Smart Beaver planned a tour around great Canadian lakes, so he asked you to help him solve the given problem.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$) — the number of integers in the sequence and the number of operations, correspondingly. The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^5$). Then follow m lines, each describes an operation. Each line starts with an integer t_i ($1 \leq t_i \leq 3$) — the operation type:

- if $t_i = 1$, then next follow two integers $x_i\ v_i$ ($1 \leq x_i \leq n, 0 \leq v_i \leq 10^5$);
- if $t_i = 2$, then next follow two integers $l_i\ r_i$ ($1 \leq l_i \leq r_i \leq n$);
- if $t_i = 3$, then next follow three integers $l_i\ r_i\ d_i$ ($1 \leq l_i \leq r_i \leq n, 0 \leq d_i \leq 10^5$).

The input limits for scoring 30 points are (subproblem E1):

- It is guaranteed that n does not exceed 100, m does not exceed 10000 and there will be no queries of the 3-rd type.

The input limits for scoring 70 points are (subproblems E1+E2):

- It is guaranteed that there will be queries of the 1-st and 2-nd type only.

The input limits for scoring 100 points are (subproblems E1+E2+E3):

- No extra limitations.

Output

For each query print the calculated sum modulo 1000000000 (10^9).

Examples

input
5 5 1 3 1 2 4 2 1 4 2 1 5 2 2 4 1 3 10 2 1 5
output
12 32 8 50

input
5 4 1 3 1 2 4 3 1 4 1 2 2 4 1 2 10 2 1 5
output
12 45

E3. Summer Homework

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

By the age of three Smart Beaver mastered all arithmetic operations and got this summer homework from the amazed teacher:

You are given a sequence of integers $a_1, a_2, ..., a_n$. Your task is to perform on it m consecutive operations of the following type:

1. For given numbers x_i and v_i assign value v_i to element a_{x_i}.
2. For given numbers l_i and r_i you've got to calculate sum $\sum_{x=l_i}^{r_i} a_x$, where $f_0 = f_1 = 1$ and at $i \geq 2$: $f_i = f_{i-1} + f_{i-2}$.
3. For a group of three numbers l_i r_i d_i you should increase value a_x by d_i for all x ($l_i \leq x \leq r_i$).

Smart Beaver planned a tour around great Canadian lakes, so he asked you to help him solve the given problem.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^5$) — the number of integers in the sequence and the number of operations, correspondingly. The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^5$). Then follow m lines, each describes an operation. Each line starts with an integer t_i ($1 \leq t_i \leq 3$) — the operation type:

- if $t_i = 1$, then next follow two integers x_i v_i ($1 \leq x_i \leq n, 0 \leq v_i \leq 10^5$);
- if $t_i = 2$, then next follow two integers l_i r_i ($1 \leq l_i \leq r_i \leq n$);
- if $t_i = 3$, then next follow three integers l_i r_i d_i ($1 \leq l_i \leq r_i \leq n, 0 \leq d_i \leq 10^5$).

The input limits for scoring 30 points are (subproblem E1):

- It is guaranteed that n does not exceed 100, m does not exceed 10000 and there will be no queries of the 3-rd type.

The input limits for scoring 70 points are (subproblems E1+E2):

- It is guaranteed that there will be queries of the 1-st and 2-nd type only.

The input limits for scoring 100 points are (subproblems E1+E2+E3):

- No extra limitations.

Output

For each query print the calculated sum modulo 1000000000 (10^9).

Examples

input
5 5 1 3 1 2 4 2 1 4 2 1 5 2 2 4 1 3 10 2 1 5
output
12 32 8 50

input
5 4 1 3 1 2 4 3 1 4 1 2 2 4 1 2 10 2 1 5
output
12 45

F1. Suns and Rays

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

Smart Beaver became interested in drawing. He draws suns. However, at some point, Smart Beaver realized that simply drawing suns is boring. So he decided to design a program that will process his drawings. You are given a picture drawn by the beaver. It will have two colors: one for the background and one for the suns in the image. Your task will be to count the number of suns in the image and for each of them to count the number of rays.

Sun is arbitrarily rotated ellipse with rays. Ray is a segment which connects point on boundary of the ellipse with some point outside ellipse.

An image where all suns are circles.

An image where all suns are ellipses, their axes are parallel to the coordinate axes.

An image where all suns are rotated ellipses.

It is guaranteed that:

- No two suns have common points.
- The rays' width is 3 pixels.
- The lengths of the ellipsis suns' axes will lie between 40 and 200 pixels.
- No two rays intersect.
- The lengths of all rays will lie between 10 and 30 pixels.

Input

The first line contains two integers h and w — the height and width of the image ($1 \le h, w \le 1600$). Next h lines will contain w space-separated integers each. They describe Smart Beaver's picture. Each number equals either a 0 (the image background), or a 1 (the sun color).

The input limits for scoring 30 points are (subproblem F1):

- All suns on the image are circles.

The input limits for scoring 70 points are (subproblems F1+F2):

- All suns on the image are ellipses with axes parallel to the coordinate axes.

The input limits for scoring 100 points are (subproblems F1+F2+F3):

- All suns on the image are ellipses, they can be arbitrarily rotated.

Output

The first line must contain a single number k — the number of suns on the beaver's image. The second line must contain exactly k space-separated integers, corresponding to the number of rays on each sun. The numbers of the second line must be **sorted** in the increasing order.

Examples

Note

For each complexity level you are suggested a sample in the initial data. You can download the samples at `http://www.abbyy.ru/sun.zip`.

F2. Suns and Rays

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

Smart Beaver became interested in drawing. He draws suns. However, at some point, Smart Beaver realized that simply drawing suns is boring. So he decided to design a program that will process his drawings. You are given a picture drawn by the beaver. It will have two colors: one for the background and one for the suns in the image. Your task will be to count the number of suns in the image and for each of them to count the number of rays.

Sun is arbitrarily rotated ellipse with rays. Ray is a segment which connects point on boundary of the ellipse with some point outside ellipse.

An image where all suns are circles.

An image where all suns are ellipses, their axes are parallel to the coordinate axes.

An image where all suns are rotated ellipses.

It is guaranteed that:

- No two suns have common points.
- The rays' width is 3 pixels.
- The lengths of the ellipsis suns' axes will lie between 40 and 200 pixels.
- No two rays intersect.
- The lengths of all rays will lie between 10 and 30 pixels.

Input

The first line contains two integers h and w — the height and width of the image ($1 \le h, w \le 1600$). Next h lines will contain w space-separated integers each. They describe Smart Beaver's picture. Each number equals either a 0 (the image background), or a 1 (the sun color).

The input limits for scoring 30 points are (subproblem F1):

- All suns on the image are circles.

The input limits for scoring 70 points are (subproblems F1+F2):

- All suns on the image are ellipses with axes parallel to the coordinate axes.

The input limits for scoring 100 points are (subproblems F1+F2+F3):

- All suns on the image are ellipses, they can be arbitrarily rotated.

Output

The first line must contain a single number k — the number of suns on the beaver's image. The second line must contain exactly k space-separated integers, corresponding to the number of rays on each sun. The numbers of the second line must be **sorted** in the increasing order.

Examples

Note

For each complexity level you are suggested a sample in the initial data. You can download the samples at `http://www.abbyy.ru/sun.zip`.

F3. Suns and Rays

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

Smart Beaver became interested in drawing. He draws suns. However, at some point, Smart Beaver realized that simply drawing suns is boring. So he decided to design a program that will process his drawings. You are given a picture drawn by the beaver. It will have two colors: one for the background and one for the suns in the image. Your task will be to count the number of suns in the image and for each of them to count the number of rays.

Sun is arbitrarily rotated ellipse with rays. Ray is a segment which connects point on boundary of the ellipse with some point outside ellipse.

An image where all suns are circles.

An image where all suns are ellipses, their axes are parallel to the coordinate axes.

An image where all suns are rotated ellipses.

It is guaranteed that:

- No two suns have common points.
- The rays' width is 3 pixels.
- The lengths of the ellipsis suns' axes will lie between 40 and 200 pixels.
- No two rays intersect.
- The lengths of all rays will lie between 10 and 30 pixels.

Input

The first line contains two integers h and w — the height and width of the image ($1 \le h, w \le 1600$). Next h lines will contain w space-separated integers each. They describe Smart Beaver's picture. Each number equals either a 0 (the image background), or a 1 (the sun color).

The input limits for scoring 30 points are (subproblem F1):

- All suns on the image are circles.

The input limits for scoring 70 points are (subproblems F1+F2):

- All suns on the image are ellipses with axes parallel to the coordinate axes.

The input limits for scoring 100 points are (subproblems F1+F2+F3):

- All suns on the image are ellipses, they can be arbitrarily rotated.

Output

The first line must contain a single number k — the number of suns on the beaver's image. The second line must contain exactly k space-separated integers, corresponding to the number of rays on each sun. The numbers of the second line must be **sorted** in the increasing order.

Examples

Note

For each complexity level you are suggested a sample in the initial data. You can download the samples at `http://www.abbyy.ru/sun.zip`.

G1. Good Substrings

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Smart Beaver recently got interested in a new word game. The point is as follows: count the number of distinct good substrings of some string s. To determine if a string is good or not the game uses rules. Overall there are n rules. Each rule is described by a group of three (p, l, r), where p is a string and l and r $(l \le r)$ are integers. We'll say that string t complies with rule (p, l, r), if the number of occurrences of string t in string p lies between l and r, inclusive. For example, string "ab", complies with rules ("ab", 1, 2) and ("aab", 0, 1), but does not comply with rules ("cd", 1, 2) and ("abab", 0, 1).

A *substring* $s[l... r]$ $(1 \le l \le r \le |s|)$ of string $s = s_1 s_2... s_{|s|}$ ($|s|$ is a length of s) is string $s_l s_{l+1}... s_r$.

Consider a *number of occurrences* of string t in string p as a number of pairs of integers l, r $(1 \le l \le r \le |p|)$ such that $p[l... r] = t$.

We'll say that string t is good if it complies with all n rules. Smart Beaver asks you to help him to write a program that can calculate the number of distinct good substrings of string s. Two substrings $s[x... y]$ and $s[z... w]$ are cosidered to be distinct iff $s[x... y] \ne s[z... w]$.

Input

The first line contains string s. The second line contains integer n. Next n lines contain the rules, one per line. Each of these lines contains a string and two integers p_i, l_i, r_i, separated by single spaces ($0 \le l_i \le r_i \le |p_i|$). It is guaranteed that all the given strings are non-empty and only contain lowercase English letters.

The input limits for scoring 30 points are (subproblem G1):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 200.

The input limits for scoring 70 points are (subproblems G1+G2):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 2000.

The input limits for scoring 100 points are (subproblems G1+G2+G3):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 50000.

Output

Print a single integer — the number of good substrings of string s.

Examples

input
aaab 2 aa 0 0 aab 1 1
output
3

input
ltntlnen 3 n 0 0 ttlneenl 1 4 lelllt 1 1
output
2

input
a 0
output
1

Note

There are three good substrings in the first sample test: «aab», «ab» and «b».

In the second test only substrings «e» and «t» are good.

G2. Good Substrings

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Smart Beaver recently got interested in a new word game. The point is as follows: count the number of distinct good substrings of some string s. To determine if a string is good or not the game uses rules. Overall there are n rules. Each rule is described by a group of three (p, l, r), where p is a string and l and r $(l \le r)$ are integers. We'll say that string t complies with rule (p, l, r), if the number of occurrences of string t in string p lies between l and r, inclusive. For example, string "ab", complies with rules ("ab", 1, 2) and ("aab", 0, 1), but does not comply with rules ("cd", 1, 2) and ("abab", 0, 1).

A *substring* $s[l... r]$ $(1 \le l \le r \le |s|)$ of string $s = s_1 s_2... s_{|s|}$ ($|s|$ is a length of s) is string $s_l s_{l+1}... s_r$.

Consider a *number of occurrences* of string t in string p as a number of pairs of integers l, r $(1 \le l \le r \le |p|)$ such that $p[l... r] = t$.

We'll say that string t is good if it complies with all n rules. Smart Beaver asks you to help him to write a program that can calculate the number of distinct good substrings of string s. Two substrings $s[x... y]$ and $s[z... w]$ are cosidered to be distinct iff $s[x... y] \ne s[z... w]$.

Input

The first line contains string s. The second line contains integer n. Next n lines contain the rules, one per line. Each of these lines contains a string and two integers p_i, l_i, r_i, separated by single spaces ($0 \le l_i \le r_i \le |p_i|$). It is guaranteed that all the given strings are non-empty and only contain lowercase English letters.

The input limits for scoring 30 points are (subproblem G1):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 200.

The input limits for scoring 70 points are (subproblems G1+G2):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 2000.

The input limits for scoring 100 points are (subproblems G1+G2+G3):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 50000.

Output

Print a single integer — the number of good substrings of string s.

Examples

input
aaab 2 aa 0 0 aab 1 1
output
3

input
ltntlnen 3 n 0 0 ttlneenl 1 4 lelllt 1 1
output
2

input
a 0
output
1

Note

There are three good substrings in the first sample test: «aab», «ab» and «b».

In the second test only substrings «e» and «t» are good.

G3. Good Substrings

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Smart Beaver recently got interested in a new word game. The point is as follows: count the number of distinct good substrings of some string s. To determine if a string is good or not the game uses rules. Overall there are n rules. Each rule is described by a group of three (p, l, r), where p is a string and l and r $(l \le r)$ are integers. We'll say that string t complies with rule (p, l, r), if the number of occurrences of string t in string p lies between l and r, inclusive. For example, string "ab", complies with rules ("ab", 1, 2) and ("aab", 0, 1), but does not comply with rules ("cd", 1, 2) and ("abab", 0, 1).

A *substring* $s[l... r]$ $(1 \le l \le r \le |s|)$ of string $s = s_1 s_2... s_{|s|}$ ($|s|$ is a length of s) is string $s_l s_{l+1}... s_r$.

Consider a *number of occurrences* of string t in string p as a number of pairs of integers l, r $(1 \le l \le r \le |p|)$ such that $p[l... r] = t$.

We'll say that string t is good if it complies with all n rules. Smart Beaver asks you to help him to write a program that can calculate the number of distinct good substrings of string s. Two substrings $s[x... y]$ and $s[z... w]$ are cosidered to be distinct iff $s[x... y] \ne s[z... w]$.

Input

The first line contains string s. The second line contains integer n. Next n lines contain the rules, one per line. Each of these lines contains a string and two integers p_i, l_i, r_i, separated by single spaces ($0 \le l_i \le r_i \le |p_i|$). It is guaranteed that all the given strings are non-empty and only contain lowercase English letters.

The input limits for scoring 30 points are (subproblem G1):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 200.

The input limits for scoring 70 points are (subproblems G1+G2):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 2000.

The input limits for scoring 100 points are (subproblems G1+G2+G3):

- $0 \le n \le 10$.
- The length of string s and the maximum length of string p is ≤ 50000.

Output

Print a single integer — the number of good substrings of string s.

Examples

input
aaab 2 aa 0 0 aab 1 1
output
3

input
ltntlnen 3 n 0 0 ttlneenl 1 4 lelllt 1 1
output
2

input
a 0
output
1

Note

There are three good substrings in the first sample test: «aab», «ab» and «b».

In the second test only substrings «e» and «t» are good.
