A. Sonya and Hotels

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

Sonya decided that having her own hotel business is the best way of earning money because she can profit and rest wherever she wants.

The country where Sonya lives is an endless line. There is a city in each integer coordinate on this line. She has n hotels, where the i-th hotel is located in the city with coordinate x_i. Sonya is a smart girl, so she does not open two or more hotels in the same city.

Sonya understands that her business needs to be expanded by opening new hotels, so she decides to build one more. She wants to make the minimum distance from this hotel to all others to be equal to d. The girl understands that there are many possible locations to construct such a hotel. Thus she wants to know the number of possible coordinates of the cities where she can build a new hotel.

Because Sonya is lounging in a jacuzzi in one of her hotels, she is asking you to find the number of cities where she can build a new hotel so that the minimum distance from the original n hotels to the new one is equal to d.

Input

The first line contains two integers n and d ($1\leq n\leq 100$, $1\leq d\leq 10^9$) — the number of Sonya's hotels and the needed minimum distance from a new hotel to all others.

The second line contains n different integers in strictly increasing order $x_1, x_2, \ldots, x_n$ ($-10^9\leq x_i\leq 10^9$) — coordinates of Sonya's hotels.

Output

Print the number of cities where Sonya can build a new hotel so that the minimum distance from this hotel to all others is equal to d.

Examples

input

```
4 3
-3 2 9 16
```

output

```
6
```

input

```
5 2
4 8 11 18 19
```

output

```
5
```

Note

In the first example, there are 6 possible cities where Sonya can build a hotel. These cities have coordinates -6, 5, 6, 12, 13, and 19.

In the second example, there are 5 possible cities where Sonya can build a hotel. These cities have coordinates 2, 6, 13, 16, and 21.

B. Sonya and Exhibition

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

Sonya decided to organize an exhibition of flowers. Since the girl likes only roses and lilies, she decided that only these two kinds of flowers should be in this exhibition.

There are n flowers in a row in the exhibition. Sonya can put either a rose or a lily in the i-th position. Thus each of n positions should contain exactly one flower: a rose or a lily.

She knows that exactly m people will visit this exhibition. The i-th visitor will visit all flowers from l_i to r_i inclusive. The girl knows that each segment has its own *beauty* that is equal to the product of the number of roses and the number of lilies.

Sonya wants her exhibition to be liked by a lot of people. That is why she wants to put the flowers in such way that the sum of *beauties* of all segments would be maximum possible.

Input

The first line contains two integers n and m ($1\leq n, m\leq 10^3$) — the number of flowers and visitors respectively.

Each of the next m lines contains two integers l_i and r_i ($1\leq l_i\leq r_i\leq n$), meaning that i-th visitor will visit all flowers from l_i to r_i inclusive.

Output

Print the string of n characters. The i-th symbol should be «`0`» if you want to put a rose in the i-th position, otherwise «`1`» if you want to put a lily.

If there are multiple answers, print any.

Examples

input

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

output

```
01100
```

input

```
6 3
5 6
1 4
4 6
```

output

```
110010
```

Note

In the first example, Sonya can put roses in the first, fourth, and fifth positions, and lilies in the second and third positions;

- in the segment $[1\ldots3]$, there are one rose and two lilies, so the *beauty* is equal to $1\cdot 2=2$;
- in the segment $[2\ldots4]$, there are one rose and two lilies, so the *beauty* is equal to $1\cdot 2=2$;
- in the segment $[2\ldots5]$, there are two roses and two lilies, so the *beauty* is equal to $2\cdot 2=4$.

The total *beauty* is equal to $2+2+4=8$.

In the second example, Sonya can put roses in the third, fourth, and sixth positions, and lilies in the first, second, and fifth positions;

- in the segment $[5\ldots6]$, there are one rose and one lily, so the *beauty* is equal to $1\cdot 1=1$;
- in the segment $[1\ldots4]$, there are two roses and two lilies, so the *beauty* is equal to $2\cdot 2=4$;
- in the segment $[4\ldots6]$, there are two roses and one lily, so the *beauty* is equal to $2\cdot 1=2$.

The total *beauty* is equal to $1+4+2=7$.

C. Sonya and Robots

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

Since Sonya is interested in robotics too, she decided to construct robots that will read and recognize numbers.

Sonya has drawn n numbers in a row, a_i is located in the i-th position. She also has put a robot at each end of the row (to the left of the first number and to the right of the last number). Sonya will give a number to each robot (they can be either same or different) and run them. When a robot is running, it is moving toward to another robot, reading numbers in the row. When a robot is reading a number that is equal to the number that was given to that robot, it will turn off and stay in the same position.

Sonya does not want robots to break, so she will give such numbers that robots will stop before they meet. That is, the girl wants them to stop at different positions so that the first robot is to the left of the second one.

For example, if the numbers $[1, 5, 4, 1, 3]$ are written, and Sonya gives the number 1 to the first robot and the number 4 to the second one, the first robot will stop in the 1-st position while the second one in the 3-rd position. In that case, robots will not meet each other. As a result, robots will not be broken. But if Sonya gives the number 4 to the first robot and the number 5 to the second one, they will meet since the first robot will stop in the 3-rd position while the second one is in the 2-nd position.

Sonya understands that it does not make sense to give a number that is not written in the row because a robot will not find this number and will meet the other robot.

Sonya is now interested in finding the number of different pairs that she can give to robots so that they will not meet. In other words, she wants to know the number of pairs (p, q), where she will give p to the first robot and q to the second one. Pairs (p_i,

q_i) and (p_j, q_j) are different if $p_i\neq p_j$ or $q_i\neq q_j$.

Unfortunately, Sonya is busy fixing robots that broke after a failed launch. That is why she is asking you to find the number of pairs that she can give to robots so that they will not meet.

Input

The first line contains a single integer n ($1\leq n\leq 10^5$) — the number of numbers in a row.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1\leq a_i\leq 10^5$) — the numbers in a row.

Output

Print one number — the number of possible pairs that Sonya can give to robots so that they will not meet.

Examples

input

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

output

```
9
```

input

```
7
1 2 1 1 1 3 2
```

output

```
7
```

Note

In the first example, Sonya can give pairs (1, 1), (1, 3), (1, 4), (1, 5), (4, 1), (4, 3), (5, 1), (5, 3), and (5, 4).

In the second example, Sonya can give pairs (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), and (3, 2).

D. Sonya and Matrix

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

Since Sonya has just learned the basics of matrices, she decided to play with them a little bit.

Sonya imagined a new type of matrices that she called *rhombic matrices*. These matrices have exactly one zero, while all other cells have the Manhattan distance to the cell containing the zero. The cells with equal numbers have the form of a rhombus, that is why Sonya called this type so.

The Manhattan distance between two cells (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$. For example, the Manhattan distance between the cells $(5, 2)$ and $(7, 1)$ equals to $|5-7|+|2-1|=3$.

5	4	3	2	3
4	3	2	1	2
3	2	1	0	1
4	3	2	1	2

Example of *a rhombic matrix*.

Note that *rhombic matrices* are uniquely defined by n, m, and the coordinates of the cell containing the zero.

She drew a $n\times m$ *rhombic matrix*. She believes that you can not recreate the matrix if she gives you only the elements of this matrix in some arbitrary order (i.e., the sequence of $n\cdot m$ numbers). Note that Sonya will not give you n and m, so only the sequence of numbers in this matrix will be at your disposal.

Write a program that finds such an $n\times m$ *rhombic matrix* whose elements are the same as the elements in the sequence in some order.

Input

The first line contains a single integer t ($1\leq t\leq 10^6$) — the number of cells in the matrix.

The second line contains t integers $a_1, a_2, \ldots, a_t$ ($0\leq a_i< t$) — the values in the cells in arbitrary order.

Output

In the first line, print two positive integers n and m ($n \times m = t$) — the size of the matrix.

In the second line, print two integers x and y ($1\leq x\leq n$, $1\leq y\leq m$) — the row number and the column number where the cell with 0 is located.

If there are multiple possible answers, print any of them. If there is no solution, print the single integer -1.

Examples

input

```
20
1 0 2 3 5 3 2 1 3 2 3 1 4 2 1 4 2 3 2 4
```

output

```
4 5
2 2
```

input

```
18
2 2 3 2 4 3 3 3 0 2 4 2 1 3 2 1 1 1
```

output

```
3 6
2 3
```

input

```
6
2 1 0 2 1 2
```

output

```
-1
```

Note

You can see the solution to the first example in the legend. You also can choose the cell $(2, 2)$ for the cell where 0 is located. You also can choose a 5×4 matrix with zero at $(4, 2)$.

In the second example, there is a 3×6 matrix, where the zero is located at $(2, 3)$ there.

In the third example, a solution does not exist.

E. Sonya and Ice Cream

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

Sonya likes ice cream very much. She eats it even during programming competitions. That is why the girl decided that she wants to open her own ice cream shops.

Sonya lives in a city with n junctions and $n-1$ streets between them. All streets are two-way and connect two junctions. It is possible to travel from any junction to any other using one or more streets. City Hall allows opening shops only on junctions. The girl cannot open shops in the middle of streets.

Sonya has exactly k friends whom she can trust. If she opens a shop, one of her friends has to work there and not to allow anybody to eat an ice cream not paying for it. Since Sonya does not want to skip an important competition, she will not work in shops personally.

Sonya wants all her ice cream shops to form a simple path of the length r ($1 \le r \le k$), i.e. to be located in different junctions $f_1, f_2, \dots, f_r$ and there is street between f_i and f_{i+1} for each i from 1 to $r-1$.

The girl takes care of potential buyers, so she also wants to minimize the maximum distance between the junctions to the nearest ice cream shop. The distance between two junctions a and b is equal to the sum of all the street lengths that you need to pass to get from the junction a to the junction b. So Sonya wants to *minimize*

$$\max_{a} \min_{1 \le i \le r} d_{a,f_i}$$

where a takes a value of all possible n junctions, f_i — the junction where the i-th Sonya's shop is located, and $d_{x,y}$ — the distance between the junctions x and y.

Sonya is not sure that she can find the optimal shops locations, that is why she is asking you to help her to open not more than k shops that will form a simple path and the maximum distance between any junction and the nearest shop would be minimal.

Input

The first line contains two integers n and k ($1\leq k\leq n\leq 10^5$) — the number of junctions and friends respectively.

Each of the next $n-1$ lines contains three integers u_i, v_i, and d_i ($1\leq u_i, v_i\leq n$, $v_i\neq u_i$, $1\leq d\leq 10^4$) — junctions that are connected by a street and the length of this street. It is guaranteed that each pair of junctions is connected by at most one street. It is guaranteed that you can get from any junctions to any other.

Output

Print one number — the minimal possible maximum distance that you need to pass to get from any junction to the nearest ice cream shop. Sonya's shops must form a simple path and the number of shops must be at most $$$k$$$.

Examples

input

```
6 2
1 2 3
2 3 4
4 5 2
4 6 3
2 4 6
```

output

```
4
```

input

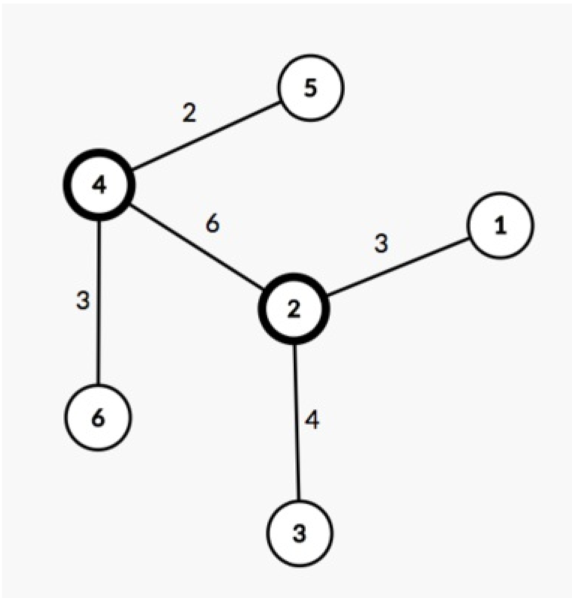
```
10 3
1 2 5
5 7 2
3 2 6
10 6 3
3 8 1
6 4 2
4 1 6
6 9 4
5 2 5
```

output

```
7
```

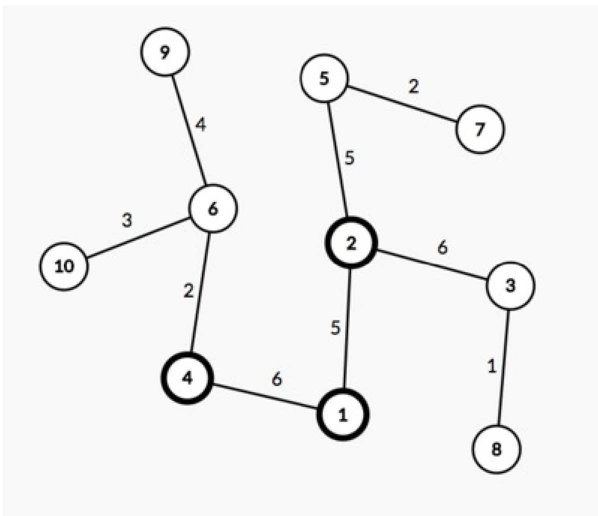
Note

In the first example, you can choose the path 2-4, so the answer will be 4.



The first example.

In the second example, you can choose the path 4-1-2, so the answer will be 7.



The second example.

F. Sonya and Bitwise OR

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Sonya has an array $a_1, a_2, \ldots, a_n$ consisting of n integers and also one non-negative integer x. She has to perform m queries of two types:

- 1 i y: replace i-th element by value y, i.e. to perform an operation a_{i} := y;
- 2 l r: find the number of pairs (L, R) that $l\leq L\leq R\leq r$ and bitwise OR of all integers in the range $[L, R]$ is at least x (note that x is a constant for all queries).

Can you help Sonya perform all her queries?

Bitwise OR is a binary operation on a pair of non-negative integers. To calculate the bitwise OR of two numbers, you need to write both numbers in binary notation. The result is a number, in binary, which contains a one in each digit if there is a one in the binary notation of at least one of the two numbers. For example, 10 `OR` 19 = 1010_2 `OR` 10011_2 = 11011_2 = 27.

Input

The first line contains three integers n, m, and x ($1\leq n, m\leq 10^5$, $0\leq x<2^{20}$) — the number of numbers, the number of queries, and the constant for all queries.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0\leq a_i<2^{20}$) — numbers of the array.

The following m lines each describe an query. A line has one of the following formats:

- 1 i y ($1\leq i\leq n$, $0\leq y<2^{20}$), meaning that you have to replace a_{i} by y;
- 2 l r ($1\leq l\leq r\leq n$), meaning that you have to find the number of subarrays on the segment from l to r that the bitwise OR of all numbers there is at least x.

Output

For each query of type 2, print the number of subarrays such that the bitwise OR of all the numbers in the range is at least x.

Examples

input

```
4 8 7
0 3 6 1
2 1 4
2 3 4
1 1 7
2 1 4
2 1 3
2 1 1
1 3 0
2 1 4
```

output

```
5
1
7
4
1
4
```

input

```
5 5 7
6 0 3 15 2
2 1 5
1 4 4
2 1 5
2 3 5
2 1 4
```

output

```
9
7
2
4
```

Note

In the first example, there are an array [0, 3, 6, 1] and queries:

1. on the segment [$1\ldots4$], you can choose pairs (1, 3), (1, 4), (2, 3), (2, 4), and (3, 4);
2. on the segment [$3\ldots4$], you can choose pair (3, 4);
3. the first number is being replacing by 7, after this operation, the array will consist of [7, 3, 6, 1];
4. on the segment [$1\ldots4$], you can choose pairs (1, 1), (1, 2), (1, 3), (1, 4),

($$$2$$$, $$$3$$$), ($$$2$$$, $$$4$$$), and ($$$3$$$, $$$4$$$);

5. on the segment [$$$1\ldots3$$$], you can choose pairs ($$$1$$$, $$$1$$$), ($$$1$$$, $$$2$$$), ($$$1$$$, $$$3$$$), and ($$$2$$$, $$$3$$$);
6. on the segment [$$$1\ldots1$$$], you can choose pair ($$$1$$$, $$$1$$$);
7. the third number is being replacing by $$$0$$$, after this operation, the array will consist of [$$$7$$$, $$$3$$$, $$$0$$$, $$$1$$$];
8. on the segment [$$$1\ldots4$$$], you can choose pairs ($$$1$$$, $$$1$$$), ($$$1$$$, $$$2$$$), ($$$1$$$, $$$3$$$), and ($$$1$$$, $$$4$$$).

In the second example, there are an array [$$$6$$$, $$$0$$$, $$$3$$$, $$$15$$$, $$$2$$$] are queries:

1. on the segment [$$$1\ldots5$$$], you can choose pairs ($$$1$$$, $$$3$$$), ($$$1$$$, $$$4$$$), ($$$1$$$, $$$5$$$), ($$$2$$$, $$$4$$$), ($$$2$$$, $$$5$$$), ($$$3$$$, $$$4$$$), ($$$3$$$, $$$5$$$), ($$$4$$$, $$$4$$$), and ($$$4$$$, $$$5$$$);
2. the fourth number is being replacing by $$$4$$$, after this operation, the array will consist of [$$$6$$$, $$$0$$$, $$$3$$$, $$$4$$$, $$$2$$$];
3. on the segment [$$$1\ldots5$$$], you can choose pairs ($$$1$$$, $$$3$$$), ($$$1$$$, $$$4$$$), ($$$1$$$, $$$5$$$), ($$$2$$$, $$$4$$$), ($$$2$$$, $$$5$$$), ($$$3$$$, $$$4$$$), and ($$$3$$$, $$$5$$$);
4. on the segment [$$$3\ldots5$$$], you can choose pairs ($$$3$$$, $$$4$$$) and ($$$3$$$, $$$5$$$);
5. on the segment [$$$1\ldots4$$$], you can choose pairs ($$$1$$$, $$$3$$$), ($$$1$$$, $$$4$$$), ($$$2$$$, $$$4$$$), and ($$$3$$$, $$$4$$$).