Codeforces Round #510 (Div. 2)

A. Benches

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

There are n benches in the Berland Central park. It is known that a_i people are currently sitting on the i-th bench. Another m people are coming to the park and each of them is going to have a seat on some bench out of n available.

Let k be the maximum number of people sitting on one bench after additional m people came to the park. Calculate the minimum possible k and the maximum possible k.

Nobody leaves the taken seat during the whole process.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of benches in the park.

The second line contains a single integer m $(1 \le m \le 10\,000)$ — the number of people additionally coming to the park.

Each of the next n lines contains a single integer a_i $(1 \le a_i \le 100)$ — the initial number of people on the i-th bench.

Output

Print the minimum possible k and the maximum possible k, where k is the maximum number of people sitting on one bench after additional m people came to the park.

Examples

input
4 6 1 1 1 1
output
3 7

input
1 10 5
output
15 15

input
3 6 1 6 5
output
6 12

input
3 7 1 6 5
output
7 13

Note

In the first example, each of four benches is occupied by a single person. The minimum k is 3. For example, it is possible to achieve if two newcomers occupy the first bench, one occupies the second bench, one occupies the third bench, and two remaining — the fourth bench. The maximum k is 7. That requires all six new people to occupy the same bench.

The second example has its minimum k equal to 15 and maximum k equal to 15, as there is just a single bench in the park and all 10 people will occupy it.

B. Vitamins

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

Berland shop sells n kinds of juices. Each juice has its price c_i. Each juice includes some set of vitamins in it. There are three types of vitamins: vitamin "A", vitamin "B" and vitamin "C". Each juice can contain one, two or all three types of vitamins in it.

Petya knows that he needs all three types of vitamins to stay healthy. What is the minimum total price of juices that Petya has to buy to obtain all three vitamins? Petya obtains some vitamin if he buys at least one juice containing it and drinks it.

Input

The first line contains a single integer n $(1 \le n \le 1\,000)$ — the number of juices.

Each of the next n lines contains an integer c_i $(1 \le c_i \le 100\,000)$ and a string s_i — the price of the i-th juice and the vitamins it contains. String s_i contains from 1 to 3 characters, and the only possible characters are "A", "B" and "C". It is guaranteed that each letter appears no more than once in each string s_i. The order of letters in strings s_i is arbitrary.

Output

Print -1 if there is no way to obtain all three vitamins. Otherwise print the minimum total price of juices that Petya has to buy to obtain all three vitamins.

Examples

input
4 5 C 6 B 16 BAC 4 A
output
15

input
2 10 AB 15 BA
output
-1

input
5 10 A 9 BC 11 CA 4 A 5 B
output
13

input
6 100 A 355 BCA 150 BC 160 AC 180 B 190 CA
output
250

input
2 5 BA 11 CB
output
16

Note
In the first example Petya buys the first, the second and the fourth juice. He spends $5 + 6 + 4 = 15$ and obtains all three vitamins. He can also buy just the third juice and obtain three vitamins, but its cost is 16, which isn't optimal.

In the second example Petya can't obtain all three vitamins, as no juice contains vitamin "C".

C. Array Product

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

You are given an array a consisting of n integers. You can perform the following operations with it:

1. Choose some positions i and j ($1 \le i, j \le n, i \ne j$), write the value of $a_i \cdot a_j$ into the j-th cell and **remove the number** from the i-th cell;
2. Choose some position i and **remove the number** from the i-th cell (this operation can be performed **no more than once and at any point of time, not necessarily in the beginning**).

The number of elements decreases by one after each operation. However, the indexing of positions stays the same. Deleted numbers can't be used in the later operations.

Your task is to perform exactly $n - 1$ operations with the array in such a way that the only number that remains in the array is maximum possible. This number can be rather large, so instead of printing it you need to print **any** sequence of operations which leads to this maximum number. Read the output format to understand what exactly you need to print.

Input
The first line contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of elements in the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the elements of the array.

Output
Print $n - 1$ lines. The k-th line should contain one of the two possible operations.

The operation of the first type should look like this: $1\ i_k\ j_k$, where 1 is the type of operation, i_k and j_k are the positions of the chosen elements.

The operation of the second type should look like this: $2\ i_k$, where 2 is the type of operation, i_k is the position of the chosen element. Note that there should be no more than one such operation.

If there are multiple possible sequences of operations leading to the maximum number — print **any** of them.

Examples

input
5 5 -2 0 1 -3
output
2 3 1 1 2 1 2 4 1 4 5

input
5 5 2 0 4 0
output
1 3 5 2 5 1 1 2 1 2 4

input
2 2 -1
output
2 2

input
4 0 -10 0 0
output
1 1 2

1 2 3
1 3 4

input
4 0 0 0 0
output
1 1 2 1 2 3 1 3 4

Note

Let X be the removed number in the array. Let's take a look at all the examples:

The first example has, for example, the following sequence of transformations of the array:
$[5, -2, 0, 1, -3] \to [5, -2, X, 1, -3] \to [X, -10, X, 1, -3] \to [X, X, X, -10, -3] \to [X, X, X, X, 30]$. Thus, the maximum answer is 30. **Note, that other sequences that lead to the answer 30 are also correct**.

The second example has, for example, the following sequence of transformations of the array:
$[5, 2, 0, 4, 0] \to [5, 2, X, 4, 0] \to [5, 2, X, 4, X] \to [X, 10, X, 4, X] \to [X, X, X, 40, X]$. The following answer is also allowed:

1 5 3
1 4 2
1 2 1
2 3
Then the sequence of transformations of the array will look like this:
$[5, 2, 0, 4, 0] \to [5, 2, 0, 4, X] \to [5, 8, 0, X, X] \to [40, X, 0, X, X] \to [40, X, X, X, X]$.

The third example can have the following sequence of transformations of the array: $[2, -1] \to [2, X]$.

The fourth example can have the following sequence of transformations of the array:
$[0, -10, 0, 0] \to [X, 0, 0, 0] \to [X, X, 0, 0] \to [X, X, X, 0]$.

The fifth example can have the following sequence of transformations of the array:
$[0, 0, 0, 0] \to [X, 0, 0, 0] \to [X, X, 0, 0] \to [X, X, X, 0]$.

D. Petya and Array

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

Petya has an array a consisting of n integers. He has learned partial sums recently, and now he can calculate the sum of elements on any segment of the array really fast. The segment is a non-empty sequence of elements standing one next to another in the array.

Now he wonders what is the number of segments in his array with the sum less than t. Help Petya to calculate this number.

More formally, you are required to calculate the number of pairs l, r ($l \le r$) such that $a_l + a_{l+1} + \dots + a_{r-1} + a_r < t$.

Input

The first line contains two integers n and t ($1 \le n \le 200\,000, |t| \le 2 \cdot 10^{14}$).

The second line contains a sequence of integers $a_1, a_2, \dots, a_n$ ($|a_i| \le 10^9$) — the description of Petya's array. Note that there might be negative, zero and positive elements.

Output

Print the number of segments in Petya's array with the sum of elements less than t.

Examples

input
5 4 5 -1 3 4 -1
output
5

input
3 0 -1 2 -3
output
4

input
4 -1 -2 1 -2 3
output
3

Note

In the first example the following segments have sum less than 4:

- $[2, 2]$, sum of elements is -1
- $[2, 3]$, sum of elements is 2
- $[3, 3]$, sum of elements is 3
- $[4, 5]$, sum of elements is 3
- $[5, 5]$, sum of elements is -1

E. Vasya and Magic Matrix

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

Vasya has got a magic matrix a of size $n \times m$. The rows of the matrix are numbered from 1 to n from top to bottom, the columns are numbered from 1 to m from left to right. Let a_{ij} be the element in the intersection of the i-th row and the j-th column.

Vasya has also got a chip. Initially, the chip is in the intersection of the r-th row and the c-th column (that is, in the element a_{rc}). Vasya performs the following process as long as possible: among all elements of the matrix having their value less than the value of the element with the chip in it, Vasya randomly and equiprobably chooses one element and moves his chip to this element.

After moving the chip, he adds to his score the square of the Euclidean distance between these elements (that is, between the element in which the chip is now and the element the chip was moved from). The process ends when there are no elements having their values less than the value of the element with the chip in it.

Euclidean distance between matrix elements with coordinates (i_1, j_1) and (i_2, j_2) is equal to $\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$.

Calculate the expected value of the Vasya's final score.

It can be shown that the answer can be represented as $\frac{P}{Q}$, where P and Q are coprime integer numbers, and $Q \not\equiv 0 \ (mod\ 998244353)$. Print the value $P \cdot Q^{-1}$ modulo 998244353.

Input

The first line of the input contains two integers n and m $(1 \le n, m \le 1\,000)$ — the number of rows and the number of columns in the matrix a.

The following n lines contain description of the matrix a. The i-th line contains m integers $a_{i1}, a_{i2}, \ldots, a_{im}$ $(0 \le a_{ij} \le 10^9)$.

The following line contains two integers r and c $(1 \le r \le n, 1 \le c \le m)$ — the index of row and the index of column where the chip is now.

Output

Print the expected value of Vasya's final score in the format described in the problem statement.

Examples

input
1 4 1 1 2 1 1 3
output
2

input
2 3 1 5 7 2 3 1 1 2
output
665496238

Note

In the first example, Vasya will move his chip exactly once. The expected value of the final score is equal to $\frac{1^2+2^2+1^2}{3} = 2$.

F. Leaf Sets

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

You are given an undirected tree, consisting of n vertices.

The vertex is called a leaf if it has exactly one vertex adjacent to it.

The distance between some pair of vertices is the number of edges in the shortest path between them.

Let's call some set of leaves *beautiful* if the maximum distance between any pair of leaves in it is less or equal to k.

You want to split **all** leaves into **non-intersecting** beautiful sets. What is the minimal number of sets in such a split?

Input

The first line contains two integers n and k ($3 \le n \le 10^6$, $1 \le k \le 10^6$) — the number of vertices in the tree and the maximum distance between any pair of leaves in each beautiful set.

Each of the next $n - 1$ lines contains two integers v_i and u_i ($1 \le v_i, u_i \le n$) — the description of the i-th edge.

It is guaranteed that the given edges form a tree.

Output

Print a single integer — the minimal number of beautiful sets the split can have.

Examples

input
9 3 1 2 1 3 2 4 2 5 3 6 6 7 6 8 3 9
output
2

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

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

Note

Here is the graph for the first example: