Problem A. George and Sleep

Time Limit	1000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

George woke up and saw the current time s on the digital clock. Besides, George knows that he has slept for time t.

Help George! Write a program that will, given time s and t, determine the time p when George went to bed. Note that George could have gone to bed yesterday relatively to the current time (see the second test sample).

Input

The first line contains current time s as a string in the format "`hh:mm`". The second line contains time t in the format "`hh:mm`" — the duration of George's sleep. It is guaranteed that the input contains the correct time in the 24-hour format, that is, $00 \le hh \le 23$, $00 \le mm \le 59$.

Output

In the single line print time p — the time George went to bed in the format similar to the format of the time in the input.

Examples

Input	Output
05:50 05:44	00:06

Input	Output
00:00 01:00	23:00

Input	Output
00:01 00:00	00:01

Note

In the first sample George went to bed at "`00:06`". Note that you should print the time only in the format "`00:06`". That's why answers "`0:06`", "`00:6`" and others will be considered incorrect.

In the second sample, George went to bed yesterday.

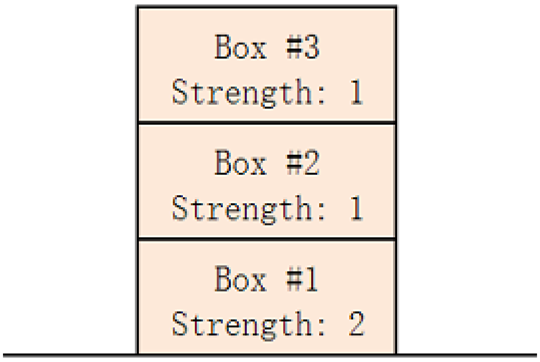
In the third sample, George didn't do to bed at all.

Problem B. Fox and Box Accumulation

Time Limit	1000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Fox Ciel has n boxes in her room. They have the same size and weight, but they might have different strength. The i-th box can hold at most x_i boxes on its top (we'll call x_i the strength of the box).

Since all the boxes have the same size, Ciel cannot put more than one box directly on the top of some box. For example, imagine Ciel has three boxes: the first has strength 2, the second has strength 1 and the third has strength 1. She cannot put the second and the third box simultaneously directly on the top of the first one. But she can put the second box directly on the top of the first one, and then the third box directly on the top of the second one. We will call such a construction of boxes a *pile*.



Fox Ciel wants to construct piles from all the boxes. Each pile will contain some boxes from top to bottom, and there cannot be more than x_i boxes on the top of i-th box. What is the minimal number of piles she needs to construct?

Input

The first line contains an integer n ($1 \le n \le 100$). The next line contains n integers $x_1, x_2, \ldots, x_n$ ($0 \le x_i \le 100$).

Output

Output a single integer — the minimal possible number of piles.

Examples

Input	Output
3 0 0 10	2

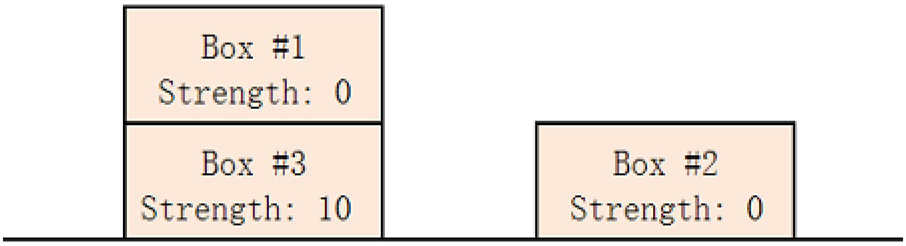
Input	Output
5 0 1 2 3 4	1

Input	Output
4 0 0 0 0	4

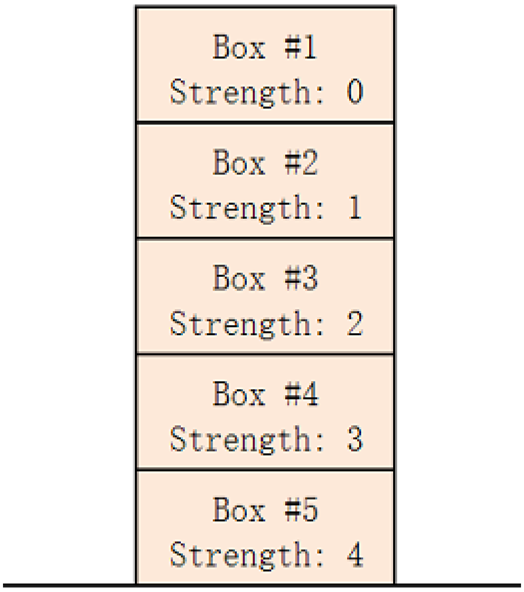
Input	Output
9 0 1 0 2 0 1 1 2 10	3

Note

In example 1, one optimal way is to build 2 piles: the first pile contains boxes 1 and 3 (from top to bottom), the second pile contains only box 2.



In example 2, we can build only 1 pile that contains boxes 1, 2, 3, 4, 5 (from top to bottom).


Box #1
Strength: 0
Box #2
Strength: 1
Box #3
Strength: 2
Box #4
Strength: 3
Box #5
Strength: 4

Problem C. Fox and Number Game

Time Limit	1000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Fox Ciel is playing a game with numbers now.

Ciel has n positive integers: $x_1, x_2, ..., x_n$. She can do the following operation as many times as needed: select two different indexes i and j such that $x_i > x_j$ hold, and then apply assignment $x_i = x_i - x_j$. The goal is to make the sum of all numbers as small as possible.

Please help Ciel to find this minimal sum.

Input

The first line contains an integer n ($2 \le n \le 100$). Then the second line contains n integers: $x_1, x_2, ..., x_n$ ($1 \le x_i \le 100$).

Output

Output a single integer — the required minimal sum.

Examples

Input	Output
2 1 2	2

Input	Output
3 2 4 6	6

Input	Output
2 12 18	12

Input	Output
5 45 12 27 30 18	15

Note

In the first example the optimal way is to do the assignment: $x_2 = x_2 - x_1$.

In the second example the optimal sequence of operations is: $x_3 = x_3 - x_2$, $x_2 = x_2 - x_1$.

Problem D. Team Olympiad

Time Limit	1000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

The School №0 of the capital of Berland has n children studying in it. All the children in this school are gifted: some of them are good at programming, some are good at maths, others are good at PE (Physical Education). Hence, for each child we know value t_i:

- $t_i = 1$, if the i-th child is good at programming,
- $t_i = 2$, if the i-th child is good at maths,
- $t_i = 3$, if the i-th child is good at PE

Each child happens to be good at exactly one of these three subjects.

The Team Scientific Decathlon Olympias requires teams of three students. The school teachers decided that the teams will be composed of three children that are good at different subjects. That is, each team must have one mathematician, one programmer and one sportsman. Of course, each child can be a member of no more than one team.

What is the maximum number of teams that the school will be able to present at the Olympiad? How should the teams be formed for that?

Input

The first line contains integer n ($1 \le n \le 5000$) — the number of children in the school. The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 3$), where t_i describes the skill of the i-th child.

Output

In the first line output integer w — the largest possible number of teams.

Then print w lines, containing three numbers in each line. Each triple represents the indexes of the children forming the team. You can print both the teams, and the numbers in the triplets in any order. The children are numbered from 1 to n in the order of their appearance in the input. Each child must participate in no more than one team. If there are several solutions, print any of them.

If no teams can be compiled, print the only line with value w equal to 0.

Examples

Input	Output
7 1 3 1 3 2 1 2	2 3 5 2 6 7 4

Input	Output
4 2 1 1 2	0

Problem E. Chip 'n Dale Rescue Rangers

Time Limit 1000 ms
Mem Limit 262144 kB

A team of furry rescue rangers was sitting idle in their hollow tree when suddenly they received a signal of distress. In a few moments they were ready, and the dirigible of the rescue chipmunks hit the road.

We assume that the action takes place on a Cartesian plane. The headquarters of the rescuers is located at point (x_1, y_1), and the distress signal came from the point (x_2, y_2).

Due to Gadget's engineering talent, the rescuers' dirigible can instantly change its current velocity and direction of movement at any moment and as many times as needed. The only limitation is: the speed of the aircraft relative to the air can not exceed $v_{\max}$ meters per second.

Of course, Gadget is a true rescuer and wants to reach the destination as soon as possible. The matter is complicated by the fact that the wind is blowing in the air and it affects the movement of the dirigible. According to the weather forecast, the wind will be defined by the vector (v_x, v_y) for the nearest t seconds, and then will change to (w_x, w_y). These vectors give both the direction and velocity of the wind. Formally, if a dirigible is located at the point (x, y), while its own velocity relative to the air is equal to zero and the wind (u_x, u_y) is blowing, then after τ seconds the new position of the dirigible will be $(x + \tau \cdot u_x, y + \tau \cdot u_y)$.

Gadget is busy piloting the aircraft, so she asked Chip to calculate how long will it take them to reach the destination if they fly optimally. He coped with the task easily, but Dale is convinced that Chip has given the random value, aiming only not to lose the face in front of Gadget. Dale has asked you to find the right answer.

It is guaranteed that the speed of the wind at any moment of time is strictly less than the maximum possible speed of the airship relative to the air.

Input

The first line of the input contains four integers x_1, y_1, x_2, y_2 ($|x_1|, |y_1|, |x_2|, |y_2| \le 10\,000$) — the coordinates of the rescuers' headquarters and the point, where signal of the distress came from, respectively.

The second line contains two integers $v_{\max}$ and t ($0 < v, t \leq 1000$), which are denoting the maximum speed of the chipmunk dirigible relative to the air and the moment of time when the wind changes according to the weather forecast, respectively.

Next follow one per line two pairs of integer (v_x, v_y) and (w_x, w_y), describing the wind for the first t seconds and the wind that will blow at all the remaining time, respectively. It is guaranteed that $v_x^2 + v_y^2 < v_{\max}^2$ and $w_x^2 + w_y^2 < v_{\max}^2$.

Output

Print a single real value — the minimum time the rescuers need to get to point (x_2, y_2). You answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$.

Examples

Input	Output
0 0 5 5 3 2 -1 -1 -1 0	3.729935587093555327

Input	Output
0 0 0 1000 100 1000 -50 0 50 0	11.547005383792516398

Problem F. Exponential notation

Time Limit 2000 ms
Mem Limit 262144 kB

You are given a positive decimal number x.

Your task is to convert it to the "`simple exponential notation`".

Let $x = a \cdot 10^b$, where $1 \le a < 10$, then in general case the "`simple exponential notation`" looks like "`aEb`". If b equals to zero, the part "`Eb`" should be skipped. If a is an integer, it should be written without decimal point. Also there should not be extra zeroes in a and b.

Input

The only line contains the positive decimal number x. The length of the line will not exceed 10^6. Note that you are given too large number, so you can't use standard built-in data types "`float`", "`double`" and other.

Output

Print the only line — the "`simple exponential notation`" of the given number x.

Examples

Input	Output
`16`	`1.6E1`

Input	Output
`01.23400`	`1.234`

Input	Output
`.100`	`1E-1`

Input	Output
`100.`	`1E2`

Problem G. Bear and Tree Jumps

Time Limit 2000 ms
Mem Limit 262144 kB

A tree is an undirected connected graph without cycles. The distance between two vertices is the number of edges in a simple path between them.

Limak is a little polar bear. He lives in a tree that consists of n vertices, numbered 1 through n.

Limak recently learned how to jump. He can jump from a vertex to any vertex within distance at most k.

For a pair of vertices (s, t) we define $f(s, t)$ as the minimum number of jumps Limak needs to get from s to t. Your task is to find the sum of $f(s, t)$ over all pairs of vertices (s, t) such that $s < t$.

Input

The first line of the input contains two integers n and k $(2 \le n \le 200\,000, 1 \le k \le 5)$ — the number of vertices in the tree and the maximum allowed jump distance respectively.

The next $n - 1$ lines describe edges in the tree. The i-th of those lines contains two integers a_i and b_i $(1 \le a_i, b_i \le n)$ — the indices on vertices connected with i-th edge.

It's guaranteed that the given edges form a tree.

Output

Print one integer, denoting the sum of $f(s, t)$ over all pairs of vertices (s, t) such that $s < t$.

Examples

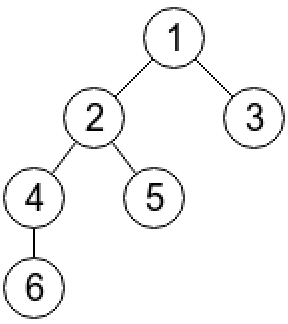
Input	Output
6 2 1 2 1 3 2 4 2 5 4 6	20

Input	Output
13 3 1 2 3 2 4 2 5 2 3 6 10 6 6 7 6 13 5 8 5 9 9 11 11 12	114

Input	Output
3 5 2 1 3 1	3

Note

In the first sample, the given tree has 6 vertices and it's displayed on the drawing below. Limak can jump to any vertex within distance at most 2. For example, from the vertex 5 he can jump to any of vertices: 1, 2 and 4 (well, he can also jump to the vertex 5 itself).



There are $\frac{n \cdot (n-1)}{2} = 15$ pairs of vertices (s, t) such that $s < t$. For 5 of those pairs Limak would need two jumps: $(1, 6)$, $(3, 4)$, $(3, 5)$, $(3, 6)$, $(5, 6)$. For other 10 pairs one jump is enough. So, the answer is $5 \cdot 2 + 10 \cdot 1 = 20$.

In the third sample, Limak can jump between every two vertices directly. There are 3 pairs of vertices $(s < t)$, so the answer is $3 \cdot 1 = 3$.