A. Lorenzo Von Matterhorn

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

Barney lives in NYC. NYC has infinite number of intersections numbered with positive integers starting from 1. There exists a bidirectional road between intersections i and $2i$ and another road between i and $2i + 1$ for every positive integer i. You can clearly see that there exists a unique shortest path between any two intersections.

Initially anyone can pass any road for free. But since SlapsGiving is ahead of us, there will q consecutive events happen soon. There are two types of events:

1. Government makes a new rule. A rule can be denoted by integers v, u and w. As the result of this action, the passing fee of all roads on the shortest path from u to v increases by w dollars.

2. Barney starts moving from some intersection v and goes to intersection u where there's a girl he wants to cuddle (using his fake name Lorenzo Von Matterhorn). He always uses the shortest path (visiting minimum number of intersections or roads) between two intersections.

Government needs your calculations. For each time Barney goes to cuddle a girl, you need to tell the government how much money he should pay (sum of passing fee of all roads he passes).

Input

The first line of input contains a single integer q ($1 \le q \le 1\,000$).

The next q lines contain the information about the events in chronological order. Each event is described in form $1\ v\ u\ w$ if it's an event when government makes a new rule about increasing the passing fee of all roads on the shortest path from u to v by w dollars, or in form $2\ v\ u$ if it's an event when Barnie goes to cuddle from the intersection v to the intersection u.

$1 \le v, u \le 10^{18}, v \ne u, 1 \le w \le 10^9$ states for every description line.

Output

For each event of second type print the sum of passing fee of all roads Barney passes in this event, in one line. Print the answers in chronological order of corresponding events.

Example

input

```
7
1 3 4 30
1 4 1 2
1 3 6 8
2 4 3
1 6 1 40
2 3 7
2 2 4
```

output

```
94
0
32
```

Note

In the example testcase:

Here are the intersections used:

1. Intersections on the path are 3, 1, 2 and 4.
2. Intersections on the path are 4, 2 and 1.
3. Intersections on the path are only 3 and 6.
4. Intersections on the path are 4, 2, 1 and 3. Passing fee of roads on the path are 32, 32 and 30 in order. So answer equals to $32 + 32 + 30 = 94$.
5. Intersections on the path are 6, 3 and 1.
6. Intersections on the path are 3 and 7. Passing fee of the road between them is 0.
7. Intersections on the path are 2 and 4. Passing fee of the road between them is 32 (increased by 30 in the first event and by 2 in the second).

B. Puzzles

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

Barney lives in country USC (United States of Charzeh). USC has n cities numbered from 1 through n and $n-1$ roads between them. Cities and roads of USC form a rooted tree (Barney's not sure why it is rooted). Root of the tree is the city number 1. Thus if one will start his journey from city 1, he can visit any city he wants by following roads.

Some girl has stolen Barney's heart, and Barney wants to find her. He starts looking for in the root of the tree and (since he is Barney Stinson not a random guy), he uses a *random DFS* to search in the cities. A pseudo code of this algorithm is as follows:

```
let starting_time be an array of length n
current_time = 0
dfs(v):
 current_time = current_time + 1
 starting_time[v] = current_time
 shuffle children[v] randomly (each permutation with equal possibility)
 // children[v] is vector of children cities of city v
 for u in children[v]:
  dfs(u)
```

As told before, Barney will start his journey in the root of the tree (equivalent to call `dfs(1)`).

Now Barney needs to pack a backpack and so he wants to know more about his upcoming journey: for every city i, Barney wants to know the expected value of `starting_time[i]`. He's a friend of Jon Snow and knows nothing, that's why he asked for your help.

Input

The first line of input contains a single integer n $(1 \le n \le 10^5)$ — the number of cities in USC.

The second line contains $n-1$ integers $p_2, p_3, ..., p_n$ $(1 \le p_i < i)$, where p_i is the number of the parent city of city number i in the tree, meaning there is a road between cities numbered p_i and i in USC.

Output

In the first and only line of output print n numbers, where i-th number is the expected value of `starting_time[i]`.

Your answer for each city will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Examples

input

```
7
1 2 1 1 4 4
```

output

```
1.0 4.0 5.0 3.5 4.5 5.0 5.0
```

input

```
12
1 1 2 2 4 4 3 3 1 10 8
```

output

```
1.0 5.0 5.5 6.5 7.5 8.0 8.0 7.0 7.5 6.5 7.5 8.0
```

C. PLEASE

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

As we all know Barney's job is "PLEASE" and he has not much to do at work. That's why he started playing "cups and key". In this game there are three identical cups arranged in a line from left to right. Initially key to Barney's heart is under the middle cup.

Then at one turn Barney swaps the cup in the middle with any of other two cups randomly (he choses each with equal probability), so the chosen cup becomes the middle one. Game lasts n turns and Barney **independently choses** a cup to swap with the middle one within each turn, and the key always remains in the cup it was at the start.

After n-th turn Barney asks a girl to guess which cup contains the key. The girl points to the middle one but Barney was distracted while making turns and doesn't know if the key is under the middle cup. That's why he asked you to tell him the probability that girl guessed right.

Number n of game turns can be extremely large, that's why Barney did not give it to you. Instead he gave you an array $a_1, a_2, ..., a_k$ such that

in other words, n is multiplication of all elements of the given array.

Because of precision difficulties, Barney asked you to tell him the answer as an irreducible fraction. In other words you need to find it as a fraction p / q such that , where is the greatest common divisor. Since p and q can be extremely large, you only need to find the remainders of dividing each of them by $10^9 + 7$.

Please note that we want of p and q to be 1, **not of their remainders** after dividing by $10^9 + 7$.

Input

The first line of input contains a single integer k $(1 \le k \le 10^5)$ — the number of elements in array Barney gave you.

The second line contains k integers $a_1, a_2, ..., a_k$ $(1 \le a_i \le 10^{18})$ — the elements of the array.

Output

In the only line of output print a single string x / y where x is the remainder of dividing p by $10^9 + 7$ and y is the remainder of dividing q by $10^9 + 7$.

Examples

input

```
1
2
```

output

```
1/2
```

input

```
3
1 1 1
```

output

```
0/1
```

D. Legen...

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Barney was hanging out with Nora for a while and now he thinks he may have feelings for her. Barney wants to send her a cheesy text message and wants to make her as happy as possible.

Initially, happiness level of Nora is 0. Nora loves some pickup lines like "I'm falling for you" and stuff. Totally, she knows n pickup lines, each consisting only of lowercase English letters, also some of them may be equal (in writing, but different in pronouncing or meaning though). Every time Nora sees i-th pickup line as a **consecutive subsequence** of Barney's text message her happiness level increases by a_i. These substrings may overlap, for example, Nora will see the pickup line `aa` twice and the pickup line `ab` once in text message `aaab`.

Due to texting app limits, Barney's text may have up to l characters.

Barney asked you to help him make Nora as much happy as possible, it's gonna be legen...

Input

The first line of input contains two integers n and l ($1 \le n \le 200$, $1 \le l \le 10^{14}$) — the number of pickup lines and the maximum length of Barney's text.

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 100$), meaning that Nora's happiness level increases by a_i after every time seeing i-th pickup line.

The next n lines contain the pickup lines. i-th of them contains a single string s_i consisting of only English lowercase letter. Summary length of all pickup lines does not exceed 200.

All strings are not empty.

Output

Print the only integer — the maximum possible value of Nora's happiness level after reading Barney's text.

Examples

input

```
3 6
3 2 1
heart
earth
art
```

output

```
6
```

input

```
3 6
3 2 8
heart
earth
art
```

output

```
16
```

Note

An optimal answer for the first sample case is `hearth` containing each pickup line exactly once.

An optimal answer for the second sample case is `artart`.

E. ...Wait for it...

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

Barney is searching for his dream girl. He lives in NYC. NYC has n junctions numbered from 1 to n and n - 1 roads connecting them. We will consider the NYC as a rooted tree with root being junction 1. m girls live in NYC, i-th of them lives along junction c_i and her weight initially equals i pounds.

Barney consider a girl x to be *better* than a girl y if and only if: girl x has weight strictly less than girl y or girl x and girl y have equal weights and index of girl x living junction index is strictly less than girl y living junction index, i.e. $c_x < c_y$. Thus for any two girls one of them is always better than another one.

For the next q days, one event happens each day. There are two types of events:

1. Barney goes from junction v to junction u. As a result he picks at most k **best girls he still have not invited** from junctions on his way and invites them to his house to test if one of them is his dream girl. If there are less than k not invited girls on his path, he invites all of them.
2. Girls living along junctions in subtree of junction v (including v itself) put on some weight. As result, their weights increase by k pounds.

Your task is for each event of first type tell Barney the indices of girls he will invite to his home in this event.

Input

The first line of input contains three integers n, m and q ($1 \le n, m, q \le 10^5$) — the number of junctions in NYC, the number of girls living in NYC and the number of events respectively.

The next n - 1 lines describes the roads. Each line contains two integers v and u ($1 \le v, u \le n, v \ne u$) meaning that there is a road connecting junctions v and u .

The next line contains m integers $c_1, c_2, ..., c_m$ ($1 \le c_i \le n$) — the girl's living junctions.

The next q lines describe the events in chronological order. Each line starts with an integer t ($1 \le t \le 2$) — type of the event .

If $t = 1$ then the line describes event of first type three integers v, u and k ($1 \le v, u, k \le n$) follow — the endpoints of Barney's path and the number of girls that he will invite at most.

Otherwise the line describes event of second type and two integers v and k ($1 \le v \le n$, $1 \le k \le 10^9$) follow — the root of the subtree and value by which all the girls' weights in the subtree should increase.

Output

For each event of the first type, print number t and then t integers $g_1, g_2, ..., g_t$ in one line, meaning that in this event Barney will invite t girls whose indices are $g_1, ..., g_t$ **in the order from the best to the worst** according to Barney's considerations.

Example

input

```
5 7 11
3 5
2 3
4 3
1 4
4 1 4 5 4 1 4
2 4 3
1 2 1 2
1 4 2 1
2 2 10
2 1 10
1 2 4 1
1 2 3 4
2 5 2
2 4 9
1 3 5 2
1 1 2 3
```

output

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

Note

For the first sample case:

Description of events:

1. Weights of girls in subtree of junction 4 increase by 3. These girls have IDs: $1, 3, 5, 4, 7$.
2. Barney goes from junction 2 to 1. Girls on his way have IDs $1, 2, 3, 5, 6, 7$ with weights $4, 2, 6, 8, 6, 10$ respectively. So, he invites girls 2 and

$1.$

3. Barney goes from junction 4 to junction 2. Girls on his way has IDs $3, 5, 7$ with weights $6, 8, 10$ respectively. So he invites girl 3.
4. Weight of girls in subtree of junction 2 increase by 10. There are no not invited girls, so nothing happens.
5. Weight of girls in subtree of junction 1 increase by 10. These girls (all girls left) have IDs: $4, 5, 6, 7$.
6. Barney goes from junction 2 to junction 4. Girls on his way has IDs $5, 7$ with weights $18, 20$ respectively. So he invites girl 5.
7. Barney goes from junction 2 to junction 3. There is no girl on his way.
8. Weight of girls in subtree of junction 5 increase by 2. The only girl there is girl with ID 4.
9. Weight of girls in subtree of junction 4 increase by 9. These girls have IDs: $4, 6, 7$.
10. Barney goes from junction 3 to junction 5. Only girl on his way is girl with ID 4.
11. Barney goes from junction 1 to junction 2. Girls on his way has IDs $6, 7$ with weights $16, 29$ respectively.

F. ...Dary!

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

Barney has finally found the one, a beautiful young lady named Lyanna. The problem is, Lyanna and Barney are trapped in Lord Loss' castle. This castle has shape of a convex polygon of n points. Like most of castles in Demonata worlds, this castle has no ceiling.

Barney and Lyanna have an escape plan, but it requires some geometry knowledge, so they asked for your help.

Barney knows that demons are organized and move in lines. He and Lyanna want to wait for the appropriate time so they need to watch for the demons. Each of them wants to stay in a point **inside the castle** (possibly on edges or corners), also they may stay in the same position. They both want to pick a real number r and watch all points in the circles with radius r around each of them (these two circles may overlap).

We say that Barney and Lyanna are *watching carefully* if and only if for every edge of the polygon, at least one of them can see at least one point on the line this edge lies on, thus such point may not be on the edge but it should be on edge's line. Formally, each edge line should have at least one common point with at least one of two circles.

The greater r is, the more energy and focus they need. So they asked you to tell them the minimum value of r such that they can watch carefully.

Input

The first line of input contains a single integer n $(3 \le n \le 300)$ — the number of castle polygon vertices.

The next n lines describe the polygon vertices in counter-clockwise order. i-th of them contains two integers x_i and y_i $(|x_i|, |y_i| \le 10^4)$ — the coordinates of i-th point of the castle. It is guaranteed that given points form a convex polygon, in particular, any three of them do not line on the same line.

Output

In the first line print the single number r — minimum radius of guys' watching circles.

In the second line print the pair of coordinates of point where Barney should stay.

In the third line print the pair of coordinates of point where Lyanna should stay.

Points should lie inside the polygon.

Coordinates may not be integers. If there are multiple answers print any of them.

Your answer will be considered correct if its absolute or relative error doesn't exceed 10^{-6}.

Examples

input

```
4
-41 67
-16 20
25 25
-36 85
```

output

```
0
-16 20
-36 85
```

input

```
7
-7 54
-5 31
-2 17
20 19
32 23
34 27
26 57
```

output

```
2.9342248
32.019503 23.0390067
-6.929116 54.006444
```

Note

In the first example guys can stay in opposite corners of the castle.