



Croatian Olympiad in Informatics

Zagreb, March 25th 2018

Tasks

Task	Time limit	Memory limit	Score
Paprike	1 second	1024 MiB	100
Pick	1 sekunda	1024 MiB	100
Svjetlost	3 seconds	1024 MiB	100
Zagonetka	3 seconds	1024 MiB	100
Total			400



Task: Paprike

Krešo went to a local family farm and bought a bunch of hot peppers that are neatly connected with pieces of string into a so-called *wreath*. In this task, a wreath consists of n peppers and $(n - 1)$ pieces of string. Each piece of string connects two different peppers, and each two peppers in the wreath are connected (directly or indirectly) by the string. Therefore, the peppers and pieces of string form a so-called *tree*. Making one cut with scissors, Krešo can cut the string, and split a pepper wreath into two smaller wreaths, which can again be split into smaller wreaths, and so on. Notice that a single pepper not connected to anything also forms a wreath.

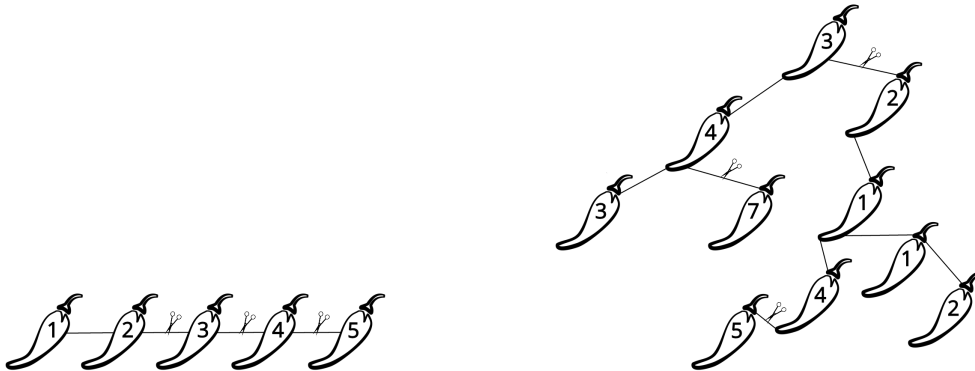


Figure 1: The initial wreaths from the first two test cases together with the optimal cuts.

The spiciness of a single pepper is measured using the so-called Scoville scale, and is represented as a non-negative integer. The spiciness of the wreath is the sum of spiciness of individual peppers it contains. Krešo wants to spice up the lunch of high school students after an informatics competition and knows that the average high school student can eat a wreath whose spiciness is at most k before they need to ask for a doctor and a juvenile lawyer.

Determine the minimal number of cuts needed so that Krešo can split the initial wreath into wreaths with spiciness at most k .

Input

The first line of input contains the integers n and k — the number of peppers and the maximal allowed spiciness of an individual wreath. The peppers are denoted with numbers from 1 to n . The following line contains n integers h_1, h_2, \dots, h_n — the number h_j is the spiciness of pepper j . Each of the following $n - 1$ lines contains two distinct integers x and y ($1 \leq x, y \leq n$) — the labels of the peppers directly connected with a piece of string in the initial wreath. The peppers and strings form a tree, as described in the task.

Output

You must output the minimal number of cuts.

Scoring

In all subtasks, it holds $n \geq 2$ i $0 \leq h_1, h_2, \dots, h_n \leq k \leq 3\,000\,000$.



Subtask	Score	Constraints
---------	-------	-------------

1	11	$n \leq 15$
2	13	$n \leq 100\,000$, each pepper $x = 1, \dots, n - 1$ is connected to pepper $x + 1$.
3	27	$n \leq 1\,000$
4	49	$n \leq 100\,000$

Sample tests

input

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

output

```
3
```

input

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

output

```
3
```

input

```
6 9
5 4 1 3 3 3
3 1
3 5
4 3
4 2
2 6
```

output

```
2
```



Task: Pick

Mirko recently read about *Pick's theorem* that says the following: in the coordinate system, if we draw a polygon whose vertices are points with integer coordinates, and if A denotes its area, i the number of points with integer coordinates located inside the polygon, and b the number of points with integer coordinates located on its edges (including the polygon's vertices), then it always holds:

$$A = i + \frac{b}{2} - 1.$$

In order to test the theorem, Mirko used his smart board to create a polygon from magnetic sticks that have, during the night, sunk to the bottom of the board because due to gravity. Now, Mirko wants to construct a polygon of the minimal possible area while using all the sticks he found. Mirko can move the sticks anywhere on his board, but he must not rotate them. Mirko is equipped with the following:

- a horizontal sticks of length 1,
- b vertical sticks of length 1,
- c diagonal sticks of length $\sqrt{2}$ that form a 45° angle with the positive part of x -axis,
- d diagonal sticks of length $\sqrt{2}$ that form a 135° angle with the positive part of x -axis.

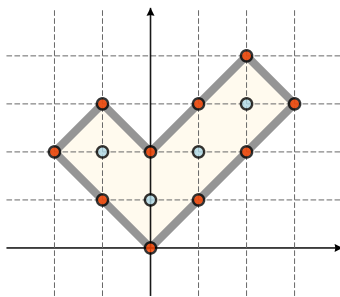


Figure 2: For the polygon above: $A = 8, i = 4, b = 10$.

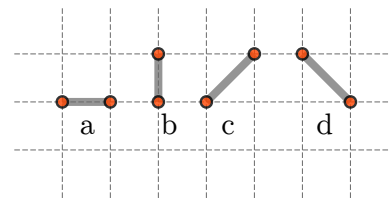


Figure 3: The sticks Mirko is equipped with.

Determine the polygon of the minimal possible area that can be obtained so that all the sticks are used. You can assume that the input data is such that it is possible to construct at least one such polygon.

Also, it is possible to score partial points if, using all of the given sticks, you construct a valid polygon (that is not necessarily of the minimal possible area). For more details, consult the section "Scoring".

Input

The first line of input contains four integers a, b, c, d from the task.

Output

You must output n lines where $n = a + b + c + d$. In the j th line, output integers x_j and y_j — the coordinates of the j th polygon vertex. The first polygon vertex must be $(0, 0)$, and the other vertices can be printed in an arbitrary direction (either positive or negative). It is allowed that the consecutive polygon sides are parallel, but the polygon cannot touch or intersect itself.



Scoring

In all subtasks, it holds $0 \leq a, b, c, d \leq 100$ and $a + b + c + d \geq 3$.

Subtask	Score	Constraints
1	5	$c = d = 0$
2	5	$a = b = 0$
3	10	$a + b + c + d \leq 6$
4	10	$a + b + c + d \leq 20$
5	10	$a + b + c + d \leq 40$
6	10	$a + b + c + d \leq 80$
7	10	$a + b + c + d \leq 150$
8	10	$a + b + c + d \leq 200$
9	10	$a + b + c + d \leq 300$
10	20	no additional constraints

If, for a test case, your solution does not output a valid polygon that consists of the given sticks, then it scores 0 points for the corresponding subtask. If the solution outputs a valid polygon that is not of the minimal possible area, then it can score partial points according to the following rules.

For test case j , let r_j denote the ratio of area of the obtained polygon and the minimal possible polygon area. For subtask k , let's denote with z_k the largest of the numbers r_j , where j is the test case from subtask k . The percentage of points P_k that the solution scores for subtask k depends on the number z_k in the following way: $P_k = 10$ if $z_k \geq 3$, and otherwise it is calculated as:

$$P_k = \frac{25}{8}(3 - z_k)^4 + 10.$$

Therefore, a solution that is not optimal can score between 10% and 60% points for a certain subtask, depending on the ratio of the area of the obtained polygon and the optimal one.

Sample tests

input

1 1 1 0

output

0 0

1 1

0 1

input

0 0 6 4

output

0 0

1 1

2 2

3 3

2 4

1 3

0 2

-1 3

-2 2

-1 1



Task: Svjetlost

In a plane, if we have a convex polygon P , and we place a source of light at a point T located outside the polygon, it lights up some edges of P — if A and B are two consecutive polygon vertices, then the edge \overline{AB} is *lit up* if the area of the triangle $\triangle TAB$ is not zero, and if it doesn't intersect the inside of the polygon. The *brightness* of the polygon is the sum of the lengths of lit up edges, and the *maximal brightness* of a polygon is the maximal possible brightness we can achieve if we select an optimal point T . The distance between point T and the polygon can be arbitrary, and the coordinates of point T don't necessarily need to be integers.

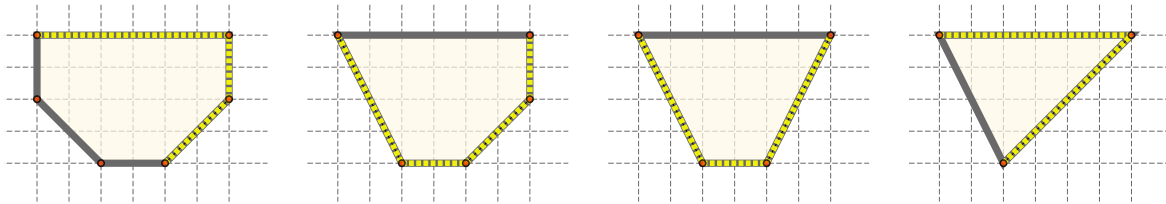


Figure 4: Polygons P , P_1 , P_2 and P_3 from the second test case, the optimal brightness is marked.

You are given a convex polygon P whose vertices are, respectively, points A_1, A_2, \dots, A_n . The polygon is changed in q steps — in the j th step, we delete an existing polygon vertex, and obtain a new polygon P_j . More precisely, the vertices of polygon P_j are the vertices of P that haven't been deleted yet, and their order is the same as in polygon P . It is easy to see that each polygon P_j is convex too.

Determine the maximal brightness of the polygon P and each of the obtained polygons P_1, P_2, \dots, P_q .

Input

The first line of input contains the positive integer n — the number of vertices of the initial polygon P . The j th of the following n lines contains two integers x_j and y_j ($-10^9 \leq x_j, y_j \leq 10^9$) — the coordinates of vertex A_j . The following line contains the integer q ($0 \leq q \leq n - 3$) — the number of steps. The j th of the following q lines contains the integer k_j ($1 \leq k_j \leq n$) that denotes that in the j th step we delete the vertex A_{k_j} . You can assume that the vertices A_j in polygon P are given counter-clockwise, that two consecutive parallel lines do not exist, and that all indices k_j are mutually distinct.

Output

You must output $q + 1$ lines. The first line must contain the maximal brightness of the initial polygon P , and the j th of the following q lines must contain the maximal brightness of polygon P_j obtained after j steps. For each line of output, an absolute and relative deviation from the official solution by 10^{-5} will be tolerated.

Scoring

Subtask	Score	Constraints
1	12	$n \leq 100$
2	14	$n \leq 2000$
3	14	$n \leq 100\,000, q = 0$
4	29	$n \leq 100\,000$, for each $j = 1, \dots, q - 1$ it holds $k_j < k_{j+1}$
5	31	$n \leq 100\,000$



Sample tests

input

4
0 0
10 0
10 10
0 10
1
2

output

20.000000
24.142136

input

6
2 2
4 0
6 0
8 2
8 4
2 4
3
1
4
3

output

10.828427
11.300563
10.944272
11.656854



Task: Zagonetka

Mislav and Marin learned about permutations in their combinatorics class, and they invented an interesting game where the player must guess certain permutations that meet certain conditions. A *permutation* of order n is an array of numbers $p = (p_1, p_2, \dots, p_n)$ where each number between 1 and n appears exactly once. A *condition* is a pair of distinct numbers (a, b) , both between 1 and n , inclusive. A permutation p *meets* condition (a, b) if $p_a < p_b$.

The game is played as follows. Marin first chooses zero or more conditions and one permutation p of order n that meets all of them. In the beginning of the game, Marin texts Mislav only the chosen permutation p (the conditions remain secret). Mislav's goal is to determine the lexicographically smallest and lexicographically largest permutation that meets all of Marin's conditions. In each step of the game, Mislav chooses one permutation q of order n and texts it to Marin. Marin then reveals if that permutation q met all of his secret conditions.

This is an interactive task. Write a program that will play the game instead of Mislav. Your program must, for a given permutation p (of length at most 100) that meets the secret conditions, in at most 5 000 steps find the lexicographically smallest and the lexicographically largest permutation that meet all the conditions.

Interaction

Before the interaction, your program must read from the standard input the following: The first line of input contains the integer n — the size of all permutations in the game. The following line contains n distinct integers p_1, p_2, \dots, p_n ($1 \leq p_j \leq n$) — the permutation p . You can assume that p meets all of Marin's conditions.

After this, your program can send Marin *queries* by writing to the standard output. Each query must be printed in its own line in the form of “**query** q_1 q_2 ... q_n ” where q_1, q_2, \dots, q_n are distinct integers between 1 and n , inclusively. After each printed query, your program must *flush* the output and read from the standard input Marin's *reply* — the number 1 if the permutation q from the query meets all the conditions, and 0 otherwise.

When your program has found a solution, it must output a line to the standard output containing the command “**end**”, then a line of the form of “ a_1 a_2 ... a_n ” containing the required lexicographically smallest permutation and a line of the form of “ b_1 b_2 ... b_n ” containing the required lexicographically largest permutation. In the end, the program must *flush* the output again and terminate the execution.

Please note: Using the evaluation system, you can obtain code samples that perform a valid interaction, including the *flush* of the output.

Interaction sample

In the following interaction sample, the first column contains the data that your program outputs to the standard output, and the second column contains the data that your program reads from the standard input. After three steps of the game, i.e. after three queries, the program determines the correct solution.



Output	Input	Please note
	4	the chosen secret conditions are (2, 1) i (3, 4)
	3 2 1 4	
query 2 3 1 4	0	condition (2, 1) is not met
query 3 2 4 1	0	condition (3, 4) is not met
query 4 1 2 3	1	both conditions are met
end		
2 1 3 4		
4 3 1 2		

Testing

You can test your solutions in two ways, locally or using the evaluation system. For both variants, you must first create an input file that contains the test case for which you want to test your program. The format of the input file must be according to the following rules: The first line of input contains the integer n — the size of all the permutations in the game. The following line contains n different integers p_1, p_2, \dots, p_n ($1 \leq p_j \leq n$) — permutation p . The following line contains the integer m ($0 \leq m \leq 10\,000$) — the number of conditions. Each of the following m lines contains two distinct integers a and b ($1 \leq a, b \leq n$) that represent a single condition. Permutation p must meet all m conditions.

For example, an input that corresponds to the above interaction sample is:

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

In order to test using the evaluation system, you first need to upload the source code of your solution using the page “Submit”, and then send the test case using the page “Test”. Local testing is done using the file “zagonetka_test” which is possible to download using the evaluation system. Testing is performed with the following command:

```
./zagonetka_test ./your_solution input_file
```

When testing using the evaluation system, as output you will get the information whether your program solved your test case correctly, and the information about the commands your program ordered, and the responses it got.

When testing locally, you will not get the information whether your program solved the test case correctly. You will need to make sure of that yourselves. The information about the course of the game will be written to the file `zagonetka.log` in the current folder.

Scoring

Subtask	Score	Constraints
1	9	$2 \leq n \leq 6$
2	18	$30 \leq n \leq 70$, Marin chose exactly 1 condition
3	22	$10 \leq n < 30$
4	51	$70 < n \leq 100$