Dress Rehearsal Problem A

Sum of Three Cubes

Time limit: 2 seconds

In 2019, a team of mathematicians found three cube numbers that sum up to 42 using over a million hours of computing time. With this breakthrough, we have found three cube numbers that sum up to all non-negative integers less than 100 that are not equal to 4 or 5 modulo 9. It has been proven in 1992 that an integer equals to 4 or 5 modulo 9 cannot be represented as a sum of three cube numbers. In other words, for each $0 \le k < 100$, we have found the triples (x, y, z) such that $x^3 + y^3 + z^3 = k$, or we have proved that no such triplet exists.

The following is a table of one of the values of (x, y, z) that satisfies $x^3 + y^3 + z^3 = k$ for each $0 \le k \le 50$ where $k \not\equiv 4 \pmod 9$ and $k \not\equiv 5 \pmod 9$.

k	x	y	z
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	1
6	−1	−1	2
7	0	−1	2
8	0	0	2
9	0	1	2
10	1	1	2
11	−2	−2	3
12	7	10	−11
15	−1	2	2
16	−511	−1609	1626
17	1	2	2
18	−1	−2	3
19	0	−2	3
20	1	−2	3
21	−11	−14	16
24	−2901096694	−15550555555	15584139827
25	−1	−1	3
26	0	−1	3
27	0	0	3
28	0	1	3
29	1	1	3
30	−283059965	−2218888517	2220422932
33	8866128975287528	−8778405442862239	−2736111468807040
34	−1	2	3
35	0	2	3
36	1	2	3
37	0	−3	4
38	1	−3	4
39	117367	134476	−159380
42	−80538738812075974	80435758145817515	12602123297335631
43	2	2	3
44	−5	−7	8
45	2	−3	4
46	−2	3	3

47	6	7	-8
48	-23	-26	31

Reading a long table is a tedious job, so you would like to create a program that takes an integer n that matches one of the values of k in the table above. Your program should produce three integers x, y, and z. The values of x, y, and z must satisfy $x^3 + y^3 + z^3 = n$, not less than -10^{18}, and not more than 10^{18}. Note that the values of x, y, and z are not required to match the table above.

Input

The first line of input contains one integer n ($0 \leq n \leq 50$, $n \not\equiv 4 \pmod 9$ and $n \not\equiv 5 \pmod 9$).

Output

Output three integers x, y, and z that satisfies $x^3 + y^3 + z^3 = n$ and $-10^{18} \leq x, y, z \leq 10^{18}$. If there is more than one solution, you can output any of them. As shown in the above table, it can be shown that at least one solution exists.

Sample Input #1	Sample Output #1
2	1214928 3480205 -3528875

Explanation for the sample input/output #1

Other answers such as $x = 3737830626090$, $y = 1490220318001$, and $z = -3815176160999$ are also accepted.

Dress Rehearsal Problem B
Symmetric Boundary
Time limit: 4 seconds

Symmetrical figures are beautiful—and they are the subject of this task. A region in a 2D plane is *convex* if, for every pair of points p and q in the region, the segment connecting p and q is entirely included in the region. Also, a region in a 2D plane is *point-symmetric* if, when you rotate the region by 180 degrees around a certain point, the rotated region exactly matches the original region.

You are given a convex polygon in a 2D plane with n vertices, numbered from 1 to n in counterclockwise order. Vertex i has coordinates (x_i, y_i). No three vertices are collinear. Determine whether there exists a convex, point-symmetric region containing all of the n vertices on its boundary. If one or more such regions exist, compute the minimum area among all of them.

Input

The first line of input contains one integer n ($3 \leq n \leq 30$). The i-th of the next m lines contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 1000$).

It is guaranteed that the given polygon is convex, its vertices are given in counterclockwise order, and no three of its vertices are collinear.

Output

If one or more such regions exist, output the minimum area among all of them. The relative error of the output must be within 10^{-9}.

If such a region does not exist, output `-1` instead.

Sample Input #1

```
4
0 0
10 0
8 9
4 9
```

Sample Output #1

```
90.0
```

Sample Input #2

```
8
8 10
2 9
0 8
0 2
2 0
8 0
10 2
10 8
```

Sample Output #2

```
-1
```

Sample Input #3	Sample Output #3
6 231 77 359 20 829 124 998 461 941 735 879 825	486567.9669655848

Explanation for the sample input/output

Figure B.1 illustrates the vertices in the sample input as black dots. For sample inputs #1 and #3, the shaded regions represent the regions with the minimum possible area.

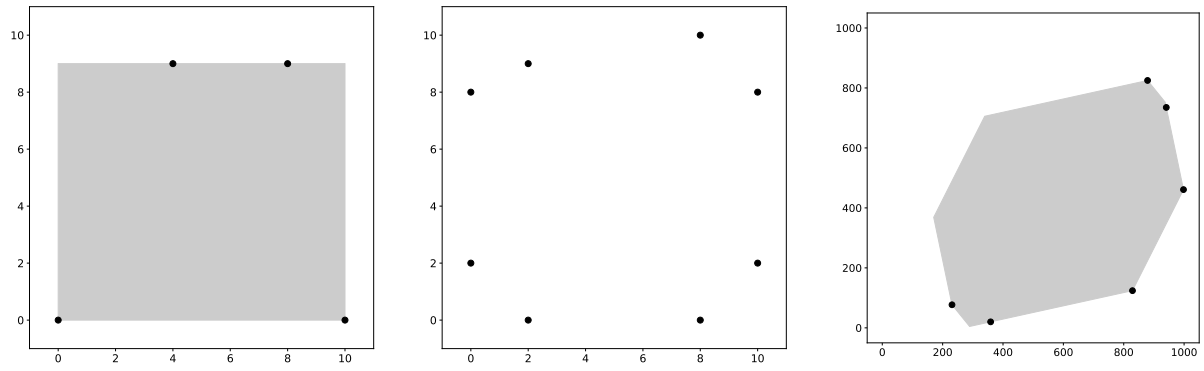


Figure B.1: Illustrations of the sample inputs (from left to right).

Dress Rehearsal Problem C
Online Assignment
Time limit: 2 seconds

You have to finish an assignment by the end of today. You can submit the assignment online and, even if your submission is rejected, you still have several more submission chances.

The assignment consists of five questions, each of which asks whether or not a given statement is true. For each submission, you have to submit answers for all five questions. If your answers are correct for all the questions, your assignment is completed. However, if any of your answers are wrong, your submission is rejected, and the number of correct answers will be told.

All the questions are so difficult that you cannot give confident answers to any of them. Your task here is to write a program that tries to submit answers repeatedly until the assignment is completed. Note that you can submit answers only a limited number of times.

Interaction

For each submission, your program should output exactly five characters without any delimiters followed by a newline to the standard output. Each character should be either `t` or `f`. The i-th character being `t` means the statement of the i-th question is judged to be true. Similarly, the character `f` means it is judged to be false.

After the output, you will receive a feedback from the standard input: either "`rejected` C" or "`completed`" followed by a newline. When you receive `rejected` C, it means that at least one of your answers is wrong, and C is the number of correct answers. In this case, your program should submit another answer. When you receive `completed`, your assignment is completed. Your program should terminate without extra output.

Your program can submit answers at most 12 times. If your 12-th submission is still rejected, your program will be judged as "Wrong Answer".

Notes on interactive judging:

- The evaluation is non-adversarial, meaning that the correct answers to the questions are chosen in advance rather than in response to your submissions.
- Do not forget to flush output buffers after writing. See the "Judging Details" document for details.
- You are provided with a command-line tool for local testing, together with input files corresponding to the sample interactions. You can download these files from DOMjudge. The tool has comments at the top to explain its use.

Read **Sample Interaction #1** **Write**

```
                                                  tffft
rejected 4
                                                  fffft
rejected 3
                                                  ttfft
completed
```

Read **Sample Interaction #2** **Write**

```
                                                  fffff
completed
```

Dress Rehearsal Problem D
Tree Quiz
Time limit: 4 seconds

Your friend wants to quiz you. You are given a *rooted tree* with n nodes, numbered from 1 to n. For every node i, its parent is node p_i, except for the *root* (the node without a parent) which has $p_i = 0$. Node u is an *ancestor* of node v if either $u = v$, or node u is an ancestor of the parent of node v (if it exists).

We say that node z is a *common ancestor* of nodes x and y if node z is an ancestor of both nodes x and y. We say that node z is the *lowest common ancestor* of nodes x and y if it is a common ancestor of nodes x and y, and every common ancestor of nodes x and y is also an ancestor of node z. We denote the lowest common ancestor of nodes x and y by $\mathrm{LCA}(x, y)$. In particular, $\mathrm{LCA}(x, x) = x$.

Your friend would like to run the following pseudocode:

```
let L be an empty array
for x = 1 to n
  for y = 1 to n
    append ((x - 1) * n * n + (LCA(x, y) - 1) * n + (y - 1)) to L
sort L in non-decreasing order
```

Your friend has q questions, numbered from 1 to q. In question j, you are given an integer k_j and asked to find the k_j-th element of the array L. Note that L is 1-indexed, so the indices range from 1 to n^2, inclusive. To pass the quiz, you have to answer all of the questions.

Input

The first line of input contains two integers n and q ($1 \le n \le 100\,000$; $1 \le q \le 100\,000$). The second line contains n integers $p_1, p_2, \ldots, p_n$ ($0 \le p_i \le n$ for all i). The input guarantees that the given values represent a rooted tree. The j-th of the next q lines contains an integer k_j ($1 \le k_j \le n^2$).

Output

For each question in order, output an integer representing the answer to the question.

Sample Input #1	Sample Output #1
5 3 3 0 2 2 3 1 18 25	0 82 124

Explanation for the sample input/output #1

The tree in the input is illustrated by Figure D.1.

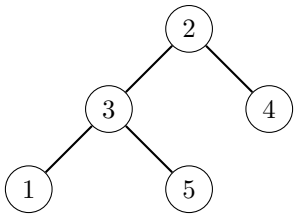


Figure D.1: Illustration of the tree in sample input #1.

The elements of L are
$(0, 6, 8, 12, 14, 30, 31, 32, 33, 34, 56, 58, 60, 62, 64, 80, 81, 82, 84, 93, 106, 108, 110, 112, 124)$.

Dress Rehearsal Problem E

Sum of Three Cubes

Time limit: 2 seconds

In 2019, a team of mathematicians found three cube numbers that sum up to 42 using over a million hours of computing time. With this breakthrough, we have found three cube numbers that sum up to all non-negative integers less than 100 that are not equal to 4 or 5 modulo 9. It has been proven in 1992 that an integer equals to 4 or 5 modulo 9 cannot be represented as a sum of three cube numbers. In other words, for each $0 \leq k < 100$, we have found the triples (x, y, z) such that $x^3 + y^3 + z^3 = k$, or we have proved that no such triplet exists.

The following is a table of one of the values of (x, y, z) that satisfies $x^3 + y^3 + z^3 = k$ for each $0 \leq k \leq 50$ where $k \not\equiv 4 \pmod 9$ and $k \not\equiv 5 \pmod 9$.

k	x	y	z
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	1
6	−1	−1	2
7	0	−1	2
8	0	0	2
9	0	1	2
10	1	1	2
11	−2	−2	3
12	7	10	−11
15	−1	2	2
16	−511	−1609	1626
17	1	2	2
18	−1	−2	3
19	0	−2	3
20	1	−2	3
21	−11	−14	16
24	−2901096694	−15550555555	15584139827
25	−1	−1	3
26	0	−1	3
27	0	0	3
28	0	1	3
29	1	1	3
30	−283059965	−2218888517	2220422932
33	8866128975287528	−8778405442862239	−2736111468807040
34	−1	2	3
35	0	2	3
36	1	2	3
37	0	−3	4
38	1	−3	4
39	117367	134476	−159380
42	−80538738812075974	80435758145817515	12602123297335631
43	2	2	3
44	−5	−7	8
45	2	−3	4
46	−2	3	3

47	6	7	-8
48	-23	-26	31

Reading a long table is a tedious job, so you would like to create a program that takes an integer n that matches one of the values of k in the table above. Your program should produce three integers x, y, and z. The values of x, y, and z must satisfy $x^3 + y^3 + z^3 = n$, not less than -10^{18}, and not more than 10^{18}. Note that the values of x, y, and z are not required to match the table above.

Input

The first line of input contains one integer n ($0 \leq n \leq 50$, $n \not\equiv 4 \pmod 9$ and $n \not\equiv 5 \pmod 9$).

Output

Output three integers x, y, and z that satisfies $x^3 + y^3 + z^3 = n$ and $-10^{18} \leq x, y, z \leq 10^{18}$. If there is more than one solution, you can output any of them. As shown in the above table, it can be shown that at least one solution exists.

Sample Input #1	Sample Output #1
2	1214928 3480205 -3528875

Explanation for the sample input/output #1

Other answers such as $x = 3737830626090$, $y = 1490220318001$, and $z = -3815176160999$ are also accepted.

Dress Rehearsal Problem F
Symmetric Boundary
Time limit: 4 seconds

Symmetrical figures are beautiful—and they are the subject of this task. A region in a 2D plane is *convex* if, for every pair of points p and q in the region, the segment connecting p and q is entirely included in the region. Also, a region in a 2D plane is *point-symmetric* if, when you rotate the region by 180 degrees around a certain point, the rotated region exactly matches the original region.

You are given a convex polygon in a 2D plane with n vertices, numbered from 1 to n in counterclockwise order. Vertex i has coordinates (x_i, y_i). No three vertices are collinear. Determine whether there exists a convex, point-symmetric region containing all of the n vertices on its boundary. If one or more such regions exist, compute the minimum area among all of them.

Input

The first line of input contains one integer n ($3 \leq n \leq 30$). The i-th of the next m lines contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 1000$).

It is guaranteed that the given polygon is convex, its vertices are given in counterclockwise order, and no three of its vertices are collinear.

Output

If one or more such regions exist, output the minimum area among all of them. The relative error of the output must be within 10^{-9}.

If such a region does not exist, output `-1` instead.

Sample Input #1

```
4
0 0
10 0
8 9
4 9
```

Sample Output #1

```
90.0
```

Sample Input #2

```
8
8 10
2 9
0 8
0 2
2 0
8 0
10 2
10 8
```

Sample Output #2

```
-1
```

Sample Input #3	Sample Output #3
6 231 77 359 20 829 124 998 461 941 735 879 825	486567.9669655848

Explanation for the sample input/output

Figure F.1 illustrates the vertices in the sample input as black dots. For sample inputs #1 and #3, the shaded regions represent the regions with the minimum possible area.

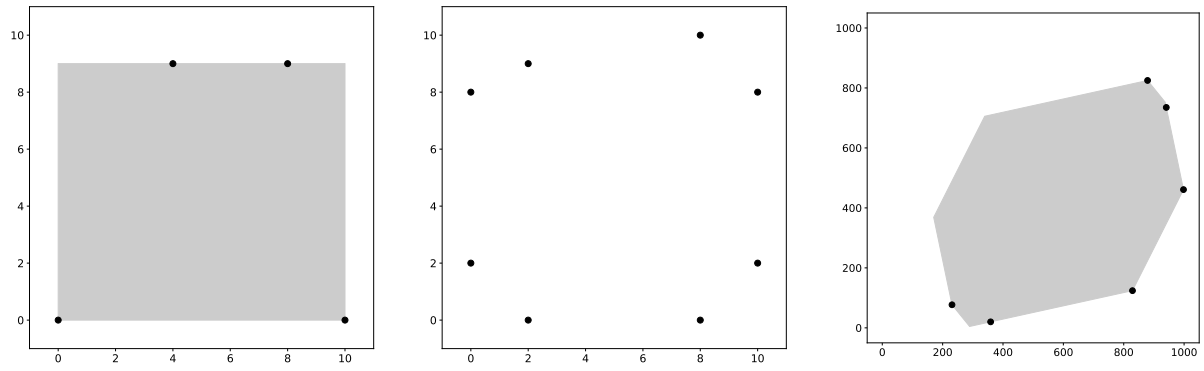


Figure F.1: Illustrations of the sample inputs (from left to right).

Dress Rehearsal Problem G
Online Assignment
Time limit: 2 seconds

You have to finish an assignment by the end of today. You can submit the assignment online and, even if your submission is rejected, you still have several more submission chances.

The assignment consists of five questions, each of which asks whether or not a given statement is true. For each submission, you have to submit answers for all five questions. If your answers are correct for all the questions, your assignment is completed. However, if any of your answers are wrong, your submission is rejected, and the number of correct answers will be told.

All the questions are so difficult that you cannot give confident answers to any of them. Your task here is to write a program that tries to submit answers repeatedly until the assignment is completed. Note that you can submit answers only a limited number of times.

Interaction

For each submission, your program should output exactly five characters without any delimiters followed by a newline to the standard output. Each character should be either `t` or `f`. The i-th character being `t` means the statement of the i-th question is judged to be true. Similarly, the character `f` means it is judged to be false.

After the output, you will receive a feedback from the standard input: either "`rejected` C" or "`completed`" followed by a newline. When you receive `rejected` C, it means that at least one of your answers is wrong, and C is the number of correct answers. In this case, your program should submit another answer. When you receive `completed`, your assignment is completed. Your program should terminate without extra output.

Your program can submit answers at most 12 times. If your 12-th submission is still rejected, your program will be judged as "Wrong Answer".

Notes on interactive judging:

- The evaluation is non-adversarial, meaning that the correct answers to the questions are chosen in advance rather than in response to your submissions.
- Do not forget to flush output buffers after writing. See the "Judging Details" document for details.
- You are provided with a command-line tool for local testing, together with input files corresponding to the sample interactions. You can download these files from DOMjudge. The tool has comments at the top to explain its use.

Read	Sample Interaction #1	Write
		tffft
rejected 4		
		fffft
rejected 3		
		ttfft
completed		

Read	Sample Interaction #2	Write
		fffff
completed		

Dress Rehearsal Problem H
Tree Quiz
Time limit: 4 seconds

Your friend wants to quiz you. You are given a *rooted tree* with n nodes, numbered from 1 to n. For every node i, its parent is node p_i, except for the *root* (the node without a parent) which has $p_i = 0$. Node u is an *ancestor* of node v if either $u = v$, or node u is an ancestor of the parent of node v (if it exists).

We say that node z is a *common ancestor* of nodes x and y if node z is an ancestor of both nodes x and y. We say that node z is the *lowest common ancestor* of nodes x and y if it is a common ancestor of nodes x and y, and every common ancestor of nodes x and y is also an ancestor of node z. We denote the lowest common ancestor of nodes x and y by $\mathrm{LCA}(x, y)$. In particular, $\mathrm{LCA}(x, x) = x$.

Your friend would like to run the following pseudocode:

```
let L be an empty array
for x = 1 to n
  for y = 1 to n
    append ((x - 1) * n * n + (LCA(x, y) - 1) * n + (y - 1)) to L
sort L in non-decreasing order
```

Your friend has q questions, numbered from 1 to q. In question j, you are given an integer k_j and asked to find the k_j-th element of the array L. Note that L is 1-indexed, so the indices range from 1 to n^2, inclusive. To pass the quiz, you have to answer all of the questions.

Input

The first line of input contains two integers n and q ($1 \le n \le 100\,000$; $1 \le q \le 100\,000$). The second line contains n integers $p_1, p_2, \ldots, p_n$ ($0 \le p_i \le n$ for all i). The input guarantees that the given values represent a rooted tree. The j-th of the next q lines contains an integer k_j ($1 \le k_j \le n^2$).

Output

For each question in order, output an integer representing the answer to the question.

Sample Input #1	Sample Output #1
5 3 3 0 2 2 3 1 18 25	0 82 124

Explanation for the sample input/output #1

The tree in the input is illustrated by Figure H.1.

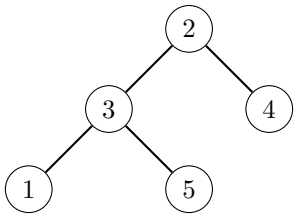


Figure H.1: Illustration of the tree in sample input #1.

The elements of L are
$(0, 6, 8, 12, 14, 30, 31, 32, 33, 34, 56, 58, 60, 62, 64, 80, 81, 82, 84, 93, 106, 108, 110, 112, 124)$.

Dress Rehearsal Problem I

Sum of Three Cubes

Time limit: 2 seconds

In 2019, a team of mathematicians found three cube numbers that sum up to 42 using over a million hours of computing time. With this breakthrough, we have found three cube numbers that sum up to all non-negative integers less than 100 that are not equal to 4 or 5 modulo 9. It has been proven in 1992 that an integer equals to 4 or 5 modulo 9 cannot be represented as a sum of three cube numbers. In other words, for each $0 \leq k < 100$, we have found the triples (x, y, z) such that $x^3 + y^3 + z^3 = k$, or we have proved that no such triplet exists.

The following is a table of one of the values of (x, y, z) that satisfies $x^3 + y^3 + z^3 = k$ for each $0 \leq k \leq 50$ where $k \not\equiv 4 \pmod 9$ and $k \not\equiv 5 \pmod 9$.

k	x	y	z
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	1
6	−1	−1	2
7	0	−1	2
8	0	0	2
9	0	1	2
10	1	1	2
11	−2	−2	3
12	7	10	−11
15	−1	2	2
16	−511	−1609	1626
17	1	2	2
18	−1	−2	3
19	0	−2	3
20	1	−2	3
21	−11	−14	16
24	−2901096694	−15550555555	15584139827
25	−1	−1	3
26	0	−1	3
27	0	0	3
28	0	1	3
29	1	1	3
30	−283059965	−2218888517	2220422932
33	8866128975287528	−8778405442862239	−2736111468807040
34	−1	2	3
35	0	2	3
36	1	2	3
37	0	−3	4
38	1	−3	4
39	117367	134476	−159380
42	−80538738812075974	80435758145817515	12602123297335631
43	2	2	3
44	−5	−7	8
45	2	−3	4
46	−2	3	3

47	6	7	-8
48	-23	-26	31

Reading a long table is a tedious job, so you would like to create a program that takes an integer n that matches one of the values of k in the table above. Your program should produce three integers x, y, and z. The values of x, y, and z must satisfy $x^3 + y^3 + z^3 = n$, not less than -10^{18}, and not more than 10^{18}. Note that the values of x, y, and z are not required to match the table above.

Input

The first line of input contains one integer n ($0 \leq n \leq 50$, $n \not\equiv 4 \pmod 9$ and $n \not\equiv 5 \pmod 9$).

Output

Output three integers x, y, and z that satisfies $x^3 + y^3 + z^3 = n$ and $-10^{18} \leq x, y, z \leq 10^{18}$. If there is more than one solution, you can output any of them. As shown in the above table, it can be shown that at least one solution exists.

Sample Input #1	Sample Output #1
2	1214928 3480205 -3528875

Explanation for the sample input/output #1

Other answers such as $x = 3737830626090$, $y = 1490220318001$, and $z = -3815176160999$ are also accepted.

Dress Rehearsal Problem J
Symmetric Boundary
Time limit: 4 seconds

Symmetrical figures are beautiful—and they are the subject of this task. A region in a 2D plane is *convex* if, for every pair of points p and q in the region, the segment connecting p and q is entirely included in the region. Also, a region in a 2D plane is *point-symmetric* if, when you rotate the region by 180 degrees around a certain point, the rotated region exactly matches the original region.

You are given a convex polygon in a 2D plane with n vertices, numbered from 1 to n in counterclockwise order. Vertex i has coordinates (x_i, y_i). No three vertices are collinear. Determine whether there exists a convex, point-symmetric region containing all of the n vertices on its boundary. If one or more such regions exist, compute the minimum area among all of them.

Input

The first line of input contains one integer n ($3 \leq n \leq 30$). The i-th of the next m lines contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 1000$).

It is guaranteed that the given polygon is convex, its vertices are given in counterclockwise order, and no three of its vertices are collinear.

Output

If one or more such regions exist, output the minimum area among all of them. The relative error of the output must be within 10^{-9}.

If such a region does not exist, output `-1` instead.

Sample Input #1

```
4
0 0
10 0
8 9
4 9
```

Sample Output #1

```
90.0
```

Sample Input #2

```
8
8 10
2 9
0 8
0 2
2 0
8 0
10 2
10 8
```

Sample Output #2

```
-1
```

Sample Input #3	Sample Output #3
6 231 77 359 20 829 124 998 461 941 735 879 825	486567.9669655848

Explanation for the sample input/output

Figure J.1 illustrates the vertices in the sample input as black dots. For sample inputs #1 and #3, the shaded regions represent the regions with the minimum possible area.

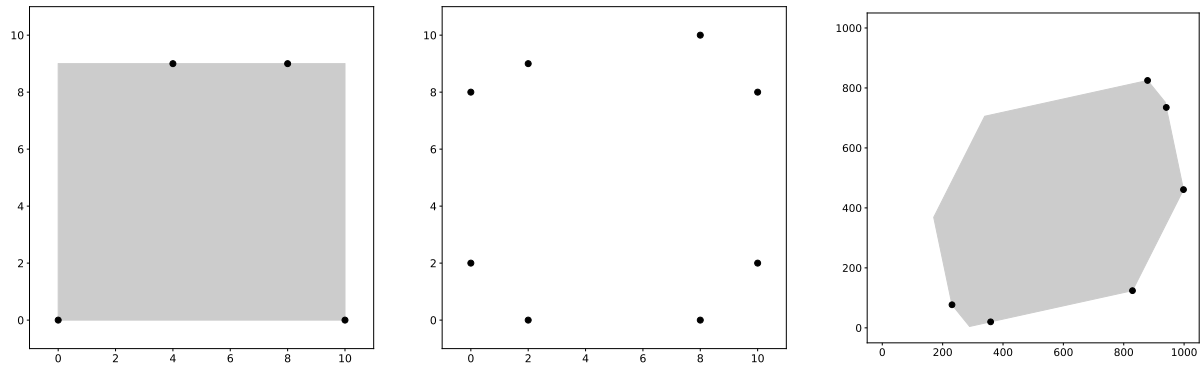


Figure J.1: Illustrations of the sample inputs (from left to right).

Dress Rehearsal Problem K
Online Assignment
Time limit: 2 seconds

You have to finish an assignment by the end of today. You can submit the assignment online and, even if your submission is rejected, you still have several more submission chances.

The assignment consists of five questions, each of which asks whether or not a given statement is true. For each submission, you have to submit answers for all five questions. If your answers are correct for all the questions, your assignment is completed. However, if any of your answers are wrong, your submission is rejected, and the number of correct answers will be told.

All the questions are so difficult that you cannot give confident answers to any of them. Your task here is to write a program that tries to submit answers repeatedly until the assignment is completed. Note that you can submit answers only a limited number of times.

Interaction

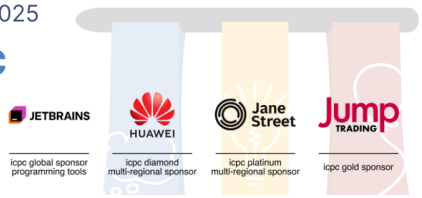
For each submission, your program should output exactly five characters without any delimiters followed by a newline to the standard output. Each character should be either `t` or `f`. The i-th character being `t` means the statement of the i-th question is judged to be true. Similarly, the character `f` means it is judged to be false.

After the output, you will receive a feedback from the standard input: either "`rejected` C" or "`completed`" followed by a newline. When you receive `rejected` C, it means that at least one of your answers is wrong, and C is the number of correct answers. In this case, your program should submit another answer. When you receive `completed`, your assignment is completed. Your program should terminate without extra output.

Your program can submit answers at most 12 times. If your 12-th submission is still rejected, your program will be judged as "Wrong Answer".

Notes on interactive judging:

- The evaluation is non-adversarial, meaning that the correct answers to the questions are chosen in advance rather than in response to your submissions.
- Do not forget to flush output buffers after writing. See the "Judging Details" document for details.
- You are provided with a command-line tool for local testing, together with input files corresponding to the sample interactions. You can download these files from DOMjudge. The tool has comments at the top to explain its use.



Read	Sample Interaction #1	Write
	`tffft`	
`rejected 4`		
	`fffft`	
`rejected 3`		
	`ttfft`	
`completed`		

Read	Sample Interaction #2	Write
	`fffff`	
`completed`		

Dress Rehearsal Problem L
Tree Quiz
Time limit: 4 seconds

Your friend wants to quiz you. You are given a *rooted tree* with n nodes, numbered from 1 to n. For every node i, its parent is node p_i, except for the *root* (the node without a parent) which has $p_i = 0$. Node u is an *ancestor* of node v if either $u = v$, or node u is an ancestor of the parent of node v (if it exists).

We say that node z is a *common ancestor* of nodes x and y if node z is an ancestor of both nodes x and y. We say that node z is the *lowest common ancestor* of nodes x and y if it is a common ancestor of nodes x and y, and every common ancestor of nodes x and y is also an ancestor of node z. We denote the lowest common ancestor of nodes x and y by $\mathrm{LCA}(x, y)$. In particular, $\mathrm{LCA}(x, x) = x$.

Your friend would like to run the following pseudocode:

```
let L be an empty array
for x = 1 to n
  for y = 1 to n
    append ((x - 1) * n * n + (LCA(x, y) - 1) * n + (y - 1)) to L
sort L in non-decreasing order
```

Your friend has q questions, numbered from 1 to q. In question j, you are given an integer k_j and asked to find the k_j-th element of the array L. Note that L is 1-indexed, so the indices range from 1 to n^2, inclusive. To pass the quiz, you have to answer all of the questions.

Input

The first line of input contains two integers n and q ($1 \le n \le 100\,000$; $1 \le q \le 100\,000$). The second line contains n integers $p_1, p_2, \ldots, p_n$ ($0 \le p_i \le n$ for all i). The input guarantees that the given values represent a rooted tree. The j-th of the next q lines contains an integer k_j ($1 \le k_j \le n^2$).

Output

For each question in order, output an integer representing the answer to the question.

Sample Input #1	Sample Output #1
5 3 3 0 2 2 3 1 18 25	0 82 124

Explanation for the sample input/output #1

The tree in the input is illustrated by Figure L.1.

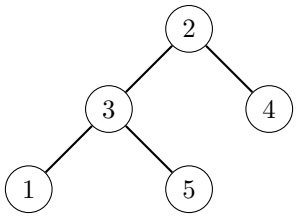


Figure L.1: Illustration of the tree in sample input #1.

The elements of L are
$(0, 6, 8, 12, 14, 30, 31, 32, 33, 34, 56, 58, 60, 62, 64, 80, 81, 82, 84, 93, 106, 108, 110, 112, 124)$.