A. New Year Table

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

Gerald is setting the New Year table. The table has the form of a circle; its radius equals R. Gerald invited many guests and is concerned whether the table has enough space for plates for all those guests. Consider all plates to be round and have the same radii that equal r. Each plate must be completely inside the table and must touch the edge of the table. Of course, the plates must not intersect, but they can touch each other. Help Gerald determine whether the table is large enough for n plates.

Input

The first line contains three integers n, R and r $(1 \le n \le 100, 1 \le r, R \le 1000)$ — the number of plates, the radius of the table and the plates' radius.

Output

Print "YES" (without the quotes) if it is possible to place n plates on the table by the rules given above. If it is impossible, print "NO".

Remember, that each plate must touch the edge of the table.

Sample test(s)

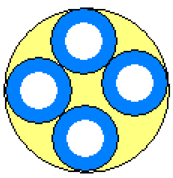
input
4 10 4
output
YES

input
5 10 4
output
NO

input
1 10 10
output
YES

Note

The possible arrangement of the plates for the first sample is:



B. New Year Cards

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

As meticulous Gerald sets the table, Alexander finished another post on Codeforces and begins to respond to New Year greetings from friends. Alexander has n friends, and each of them sends to Alexander exactly one e-card. Let us number his friends by numbers from 1 to n in the order in which they send the cards. Let's introduce the same numbering for the cards, that is, according to the numbering the i-th friend sent to Alexander a card number i.

Alexander also sends cards to friends, but he doesn't look for the new cards on the Net. He simply uses the cards previously sent to him (sometimes, however, he does need to add some crucial details). Initially Alexander doesn't have any cards. Alexander always follows the two rules:

1. He will never send to a firend a card that this friend has sent to him.
2. Among the other cards available to him at the moment, Alexander always chooses one that Alexander himself likes most.

Alexander plans to send to each friend exactly one card. Of course, Alexander can send the same card multiple times.

Alexander and each his friend has the list of preferences, which is a permutation of integers from 1 to n. The first number in the list is the number of the favorite card, the second number shows the second favorite, and so on, the last number shows the least favorite card.

Your task is to find a schedule of sending cards for Alexander. Determine at which moments of time Alexander must send cards to his friends, to please each of them as much as possible. In other words, so that as a result of applying two Alexander's rules, each friend receives the card that is preferred for him as much as possible.

Note that Alexander doesn't choose freely what card to send, but he always strictly follows the two rules.

Input

The first line contains an integer n $(2 \le n \le 300)$ — the number of Alexander's friends, equal to the number of cards. Next n lines contain his friends' preference lists. Each list consists of n different integers from 1 to n. The last line contains Alexander's preference list in the same format.

Output

Print n space-separated numbers: the i-th number should be the number of the friend, whose card Alexander receives right before he should send a card to the i-th friend. If there are several solutions, print any of them.

Sample test(s)

input

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

output

```
2 1 1 4
```

Note

In the sample, the algorithm of actions Alexander and his friends perform is as follows:

1. Alexander receives card 1 from the first friend.
2. Alexander sends the card he has received (at the moment he only has one card, and therefore it is the most preferable for him) to friends with the numbers 2 and 3.
3. Alexander receives card 2 from the second friend, now he has two cards — 1 and 2.
4. Alexander sends a card to the first friend. Despite the fact that Alexander likes card 1 more, he sends card 2 as he cannot send a friend the card sent by that very friend.
5. Alexander receives card 3 from the third friend.
6. Alexander receives card 4 from the fourth friend.
7. Among the cards Alexander has number 3 is his favorite and he sends it to the fourth friend.

Note that Alexander can send cards to multiple friends at a time (in this case the second and the third one). Alexander can send card 3 to the fourth friend after he receives the third card or after he receives the fourth card (both variants are correct).

C. New Year Snowmen

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

As meticulous Gerald sets the table and caring Alexander sends the postcards, Sergey makes snowmen. Each showman should consist of three snowballs: a big one, a medium one and a small one. Sergey's twins help him: they've already made n snowballs with radii equal to r_1, r_2, ..., r_n. To make a snowman, one needs any three snowballs whose radii are pairwise different. For example, the balls with radii 1, 2 and 3 can be used to make a snowman but $2, 2, 3$ or $2, 2, 2$ cannot. Help Sergey and his twins to determine what **maximum** number of snowmen they can make from those snowballs.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of snowballs. The next line contains n integers — the balls' radii r_1, r_2, ..., r_n ($1 \le r_i \le 10^9$). The balls' radii can coincide.

Output

Print on the first line a single number k — the maximum number of the snowmen. Next k lines should contain the snowmen's descriptions. The description of each snowman should consist of three space-separated numbers — the big ball's radius, the medium ball's radius and the small ball's radius. It is allowed to print the snowmen in any order. If there are several solutions, print any of them.

Sample test(s)

input

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

output

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

input

```
3
2 2 3
```

output

```
0
```

D. New Year Contest

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

As Gerald sets the table, Alexander sends the greeting cards, and Sergey and his twins create an army of clone snowmen, Gennady writes a New Year contest.

The New Year contest begins at 18:00 (6.00 P.M.) on December 31 and ends at 6:00 (6.00 A.M.) on January 1. There are n problems for the contest. The penalty time for each solved problem is set as the distance from the moment of solution submission to the New Year in minutes. For example, the problem submitted at 21:00 (9.00 P.M.) gets penalty time 180, as well as the problem submitted at 3:00 (3.00 A.M.). The total penalty time is calculated as the sum of penalty time for all solved problems. It is allowed to submit a problem exactly at the end of the contest, at 6:00 (6.00 A.M.).

Gennady opened the problems exactly at 18:00 (6.00 P.M.) and managed to estimate their complexity during the first 10 minutes of the contest. He believes that writing a solution for the i-th problem will take a_i minutes. Gennady can submit a solution for evaluation at any time after he completes writing it. Probably he will have to distract from writing some solution to send the solutions of other problems for evaluation. The time needed to send the solutions can be neglected, i.e. this time can be considered to equal zero. Gennady can simultaneously submit multiple solutions. Besides, he can move at any time from writing one problem to another, and then return to the first problem from the very same place, where he has left it. Thus the total solution writing time of the i-th problem always equals a_i minutes. Of course, Gennady does not commit wrong attempts, and his solutions are always correct and are accepted from the first attempt. He can begin to write the solutions starting from 18:10 (6.10 P.M.).

Help Gennady choose from the strategies that help him solve the maximum possible number of problems, the one with which his total penalty time will be minimum.

Input

The first line contains an integer n $(1 \le n \le 100)$ — the number of the problems. The next line contains n space-separated integers a_i $(1 \le a_i \le 720)$ — each number shows how much time in minutes Gennady will spend writing a solution to the problem.

Output

Print two integers — the number of problems Gennady will solve and the total penalty time considering that he chooses the optimal strategy.

Sample test(s)

input

```
3
30 330 720
```

output

```
2 10
```

Note

In the sample, one of Gennady's possible optimal strategies is as follows. At 18:10 (6:10 PM) he begins to write the first problem and solves it in 30 minutes (18:40 or 6.40 P.M.). At 18:40 (6.40 P.M.) he begins to write the second problem. There are 320 minutes left before the New Year, so Gennady does not have the time to finish writing the second problem before the New Year. At 0:00 (12.00 A.M.) he distracts from the second problem, submits the first one, and returns immediately to writing the second problem. At 0:10 (0.10 A.M.), he completes the solution for the second problem, submits it and gets 10 minute penalty time. Note that as the total duration of the contest is 720 minutes and Gennady has already spent 10 minutes on reading the problems, he will not have time to solve the third problem during the contest. Yes, such problems happen to exist.

Competitions by the given rules are held annually on the site http://b23.ru/3wvc

E. New Year Garland

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

As Gerald, Alexander, Sergey and Gennady are already busy with the usual New Year chores, Edward hastily decorates the New Year Tree. And any decent New Year Tree must be decorated with a good garland. Edward has lamps of m colors and he wants to make a garland from them. That garland should represent a sequence whose length equals L. Edward's tree is n layers high and Edward plans to hang the garland so as to decorate the first layer with the first l_1 lamps, the second layer — with the next l_2 lamps and so on. The last n-th layer should be decorated with the last l_n lamps, $\sum_{i=1}^{n} l_i = L$.

Edward adores all sorts of math puzzles, so he suddenly wondered: how many different ways to assemble the garland are there given that the both following two conditions are met:

1. Any two lamps that follow consecutively in the same layer should have different colors.
2. The sets of used colors in every two **neighbouring** layers must be different. We consider unordered sets (not multisets), where every color occurs no more than once. So the number of lamps of particular color does not matter.

Help Edward find the answer to this nagging problem or else he won't manage to decorate the Tree by New Year. You may consider that Edward has an unlimited number of lamps of each of m colors and it is not obligatory to use all m colors. The garlands are considered different if they differ in at least one position when represented as sequences. Calculate the answer modulo p.

Input

The first line contains three integers n, m and p ($1 \le n, m \le 10^6$, $2 \le p \le 10^9$) which are the number of the tree's layers, the number of the lamps' colors and module correspondingly. The next line contains n integers l_i ($1 \le l_i \le 5000$, $L = \sum_{i=1}^{n} l_i \le 10^7$).

Output

Print the only integer — the number of garlands modulo p.

Sample test(s)

input

```
3 2 1000
3 1 2
```

output

```
8
```

input

```
2 3 1000
2 2
```

output

```
24
```

input

```
1 1 1000
5
```

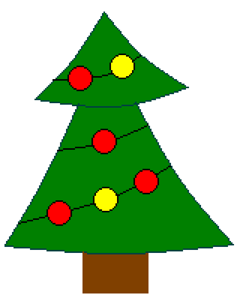
output

```
0
```

Note

In the first sample the following variants are possible: 121|1|12, 121|1|21, 121|2|12, 121|2|21, 212|1|12, 212|1|21, 212|2|12, 212|2|21. In the second sample the following variants are possible: 12|13, 12|23, 12|31, 12|32 and so on.

Figure for the first sample:



F. New Year Snowflake

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

As Gerald ..., in other words, on a New Year Eve Constantine prepared an unusual present for the Beautiful Lady. The present is the magic New Year snowflake that can make any dream come true.

The New Year snowflake consists of tiny ice crystals, which can be approximately regarded as points on the plane. The beauty of the New Year snowflake is that it has a center of symmetry. This is a point such that for each crystal of the snowflake exists another crystal, symmetrical to it relative to that point. One of the crystals can be placed directly in the center of symmetry.

While Constantine was choosing a snowflake among millions of other snowflakes, no less symmetrical and no less magical, then endured a difficult path through the drifts to the house of his mistress, while he was waiting with bated breath for a few long moments before the Beautiful Lady opens the door, some of the snowflake crystals melted and naturally disappeared. Constantine is sure that there were no more than k of such crystals, because he handled the snowflake very carefully. Now he is ready to demonstrate to the Beautiful Lady all the power of nanotechnology and restore the symmetry of snowflakes.

You are given the coordinates of the surviving snowflake crystals, given in nanometers. Your task is to identify all possible positions of the original center of symmetry.

Input

The first line contains two integers n and k ($1 \le n \le 200\,000$, $0 \le k \le 10$) — the number of the surviving snowflake crystals and the maximum number of melted crystals, correspondingly. Next n lines contain the coordinates of the crystals that are left in the following form: "x_i y_i". The coordinates are integers and do not exceed $5 \cdot 10^8$ in absolute value. All given points are different.

Output

The first line contains an integer c — the number of possible symmetry centers. Next c lines should contain the centers' descriptions. Each symmetry center is described by a couple of coordinates "x y", separated by a space. Print the coordinates with absolute error not exceeding 10^{-6}. You are allowed to print the symmetry centers in any order. All printed points should be different. If there exist an infinite number of possible symmetry centers, print the single number "-1".

Sample test(s)

input

```
4 0
0 0
0 1
1 0
1 1
```

output

```
1
0.5 0.5
```

input

```
4 2
0 0
0 1
1 0
1 1
```

output

```
5
0.0 0.5
0.5 0.0
0.5 0.5
0.5 1.0
1.0 0.5
```

input

```
4 4
0 0
0 1
1 0
1 1
```

output

```
-1
```