Problem G. Nano alarm-clocks

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

An old watchmaker has n stopped nano alarm-clocks numbered with integers from 1 to n. Nano alarm-clocks count time in hours, and in one hour there are million minutes, each minute lasting a million seconds. In order to repair them all the watchmaker should synchronize the time on all nano alarm-clocks. In order to do this he moves clock hands a certain time forward (may be zero time). Let's name this time shift a transfer time.

Your task is to calculate the minimal total transfer time required for all nano alarm-clocks to show the same time.

Input

The first line contains a single integer n — the number of nano alarm-clocks ($2 \le n \le 10^5$). In each i-th of the next n lines the time h, m, s, shown on the i-th clock. Integers h, m and s show the number of hours, minutes and seconds respectively. ($0 \le h < 12$, $0 \le m < 10^6$, $0 \le s < 10^6$).

Output

Output three integers separated with spaces h, m and s — total minimal transfer time, where h, m and s — number of hours, minutes and seconds respectively ($0 \le m < 10^6$, $0 \le s < 10^6$).

Examples

stdin	stdout

2
10 0 0
3 0 0
	5 0 0

3
11 999999 999999
0 0 0
11 999999 999999
	0 0 2

Problem H. Lunch

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

The swamp looks like a narrow lane with length n covered by floating leaves sized 1, numbered from 1 to n with a fly sitting on the top of each. A little toad is sitting on one of the leaves instead of a fly. Its name is Kvait and it is about to have lunch. It can jump to the bordering leaf or jump it over to the next one in any direction. When landing it eats a fly. Kvait is already quite a big toad and the leaves are unstable so when it jumps away the leaf starts sinking.

In order to have lunch Kvait needs to eat all of the flies. It starts his journey from the leaf with number s and has to finish on the leaf with number f. Yet jumping to the bordering leaf takes more Kvait's energy than skipping a leaf over. It is necessary to plan the toad's movements to get lunch with minimal energy spent.

Input

Single line contains three integers n, s, f ($2 \le n \le 10\,000$, $1 \le s, f \le n$) — the number of leaves, number of a starting leaf and the number of the finish leaf respectively.

Output

Output the minimal number of jumps to the bordering leaves required for the toad to have lunch. If there is no way to eat up, output a single number -1.

Examples

stdin	stdout
`4 1 2`	`1`

Problem I. Accounting Numeral System

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

The latest Accounting Numeral System is the top accounting system in the whole world. Its creator, Dr. Ceizenpok, is the best expert of the respective authority. Any positive integer n in this system based m is represented as a sum of m parts:

$$n = C_{x_m}^{m} + C_{x_{m-1}}^{m-1} + C_{x_{m-2}}^{m-2} + \ldots + C_{x_1}^{1},$$

while $x_1, x_2, \ldots, x_m$ — are such integers that $0 \le x_1 < x_2 < \ldots < x_m$. Numbers $C_k^m = \frac{k!}{m!\,(k-m)!}$ our experts call accounting indexes. Each number n in this system is recorded as $n = \overline{(x_m)\ldots(x_2)(x_1)}$, and it is considered that $0! = 1$ and $C_k^m = 0$, if $m > k$. For example, number 9 in the accounting system based 3 is recorded as $(\mathbf{4})(\mathbf{3})(\mathbf{2})$, because $9 = C_{\mathbf{4}}^3 + C_{\mathbf{3}}^2 + C_{\mathbf{2}}^1$, and number 1 in this system based 2 looks like:$(\mathbf{2})(\mathbf{0})$, because $1 = C_{\mathbf{2}}^2 + C_{\mathbf{0}}^1$.

You have to find a representation of an integer n in the accounting numeral system based m.

Input

Single line contains two integers n and m ($1 \le n \le 10^{16}$, $2 \le m \le 1\,000$).

Output

Single line should contain a sequence of m space-separated integers $x_m, \ldots, x_2, x_1$, that form a number designation n in the accounting numeral system. Number x_m is the leftmost digit in the number designation n, and x_1 — its rightmost one.

Examples

stdin	stdout
`9 3`	`4 3 2`
`5 2`	`3 2`

Problem J. Ceizenpok's formula

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

Dr. Ceizenpok from planet i1c5l became famous across the whole Universe thanks to his recent discovery — the Ceizenpok's formula. This formula has only three arguments: n, k and m, and its value is a number of k-combinations of a set of n modulo m.

While the whole Universe is trying to guess what the formula is useful for, we need to automate its calculation.

Input

Single line contains three integers n, k, m, separated with spaces ($1 \le n \le 10^{18}$, $0 \le k \le n$, $2 \le m \le 1\,000\,000$).

Output

Write the formula value for given arguments n, k, m.

Examples

stdin	stdout
`2 1 3`	`2`
`4 2 5`	`1`

Problem K. Dividing an orange

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

On a far-away planet i1c5l a community of n people harvested k oranges. Now they have to divide the harvest among themselves.

This community is ruled by a democratic principle based on the rank hierarchy. Thus, the harvest is divided the following way: each person gets a rank from 1 to n. Then, a person ranked 1 announce his decision: who and how many oranges get. Afterwards all n people vote «for» or «against». If at least half of the people vote «for», then the decision is accepted. Otherwise the person ranked 1 is ostracized from the community and it is turn to person ranked 2 to announce a decision, and the procedure is repeated.

Each person wish to get the best for himself trying to get as many oranges as possible. Between the cases with equal amount of oranges earned he will prefer the one with less people in community left. If a person is ostracized from the community it is considered that he got a negative amount of oranges. If several optimal solutions exist a person can choose any. Each person knows that other people also try to find the optimal solution being guided by the same principles.

However, one of the community members has m wildcards that can give him predefined ranks. The task is to find out the minimal and the maximum number of oranges that can be obtained for each wildcard rank.

Input

The first line contains integers n, k and m ($1 \le n,\ k \le 10^9$, $1 \le m \le 10^5$) — the amount of people, oranges and wildcards respectively.

The second line contains m integers a_1, a_2, ..., a_m ($1 \le ai \le n$), where a_i — the rank given by i-th wildcard.

Output

For each of a_i write a minimal and a maximum amount of oranges on a separate line, which will be obtained by the wildcard owner or «-1 -1» (without quote), if he or she will be ostracized.

Examples

stdin	stdout

`2 10 2`
`1 2`
	`10 10`
	`0 0`

`3 1 2`
`1 3`
	`0 0`
	`1 1`

Note

In the first example the person with the first rank takes all the oranges and votes «for».

In the second example the person ranked 1 has to give the orange to the person of third rank. If the first one takes one orange then the rest of the members will vote «against». If the first gives the orange to the second then the second will also be «against» as he knows that by ostracizing the first they will also get an orange but there will be less people left in the community. Because the third one will also be «against» this is not an option for the first one.

Problem L. The Pool for Lucky Ones

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

A new swimming pool has been built in Kazan for the forthcoming Water Sports World Championship. The pool has N lanes. Some of the lanes are already occupied by swimmers. Tatar scientists have divided the lanes into the lucky and unlucky ones. The unlucky lanes are those with the maximum amount of swimmers. That is, there is no other lane where there would be more swimmers than on unlucky one. The unlucky lanes make swimmers unhappy. The rest of the lanes are considered to be lucky. The lucky lanes make people happy. The scientists took a decision to make more people happy. In order to do this they had an agreement with the pool manager saying they can move a single person from any lane to the one neighboring if it was necessary. The swimmer from the first lane can only be moved to the second lane, and the swimmer from the last lane — to the one before last.

Input

The first line contains an integer N — the amount of lanes in the pool ($3 \le N \le 10^5$). The second line contains N integers p_i separated with spaces, describing distribution of swimmers between the lanes where p_i is the amount of swimmers on i-th lane ($0 \le p_i \le 10^5$).

Output

Output a single number — minimal possible number of unhappy swimmers.

Examples

standard input	standard output

`3`
`1 3 5`
	`5`

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

Problem M. #TheDress

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After landing on planet i1c5l people noticed that blue and black clothes are quite popular among the locals. Each aboriginal has at least one blue-and-black piece of clothing in their wardrobe. This makes no interest except one curious detail: the locals claimed that these colors weren't blue and black but white and gold.

Thus a simple test was created to differ a human being from an alien. On one of the wedding parties people took a picture of the blue-and-black groom mother's dress. This picture was shown to some respondents who were asked the color of the dress. If the answer contained «blue» and «black» then there was no doubt that the respondent was from the Earth. The answer containing «white» and «gold» pointed to the person of planet i1c5l origin. If the answer contained neither of word pairs then it was clear that the respondent was a creature from another planet.

You have the complete survey log from planet i1c5l. Your task is to determine the constitution of the planet's population based on the survey.

Input

The first line contains single integer N — the number of respondents ($1 \le N \le 100$). The following N lines contain the answers. No line is empty and no line is longer than 100 characters. The answer contains only lower-case Latin letters and spaces. It is guaranteed that no answer can contain «blue», «black», «white», and «gold» simultaneously.

Output

Output three numbers that describe the planet's population, each on separate line.

The first number — percentage of earthlings in population.

The second number — percentage of aboriginals in population.

The third number — percentage of another planet creatures in population.

Output all numbers with 10^{-5} accuracy.

Examples

standard input	standard output

`3`
`goldandwhite`
`white and pinkman`
`blueblueblue and a little bit black`
	`33.3333333333`
	`33.3333333333`
	`33.3333333333`

`4`
`this dress is blue and black`
`this dress is goldblackblue`
`eto plate kazhetsia sirenevenkoe`
`no comments`
	`50.0000000000`
	`0.0000000000`
	`50.0000000000`

Problem N. A frog in the desert

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

In the endless desert at point A there is a strategic nuclear submarine "The Frog" buried in the sand. The military headquarters decided to change the dislocation of The Frog to point B. Without having any chance to cancel that decision it came an issue to find out how the buried submarine could be moved from one point of the flat desert to another.

It turned out that The Frog's design engineering team solved that problem years ago during construction stage. Design engineers installed the real teleport on The Frog. The problem seemed over, as it turned out that the teleport has only K operation modes which differ only in the teleportation distance, that is for each mode there is a predefined distance L_i, which the submarine would move in a given direction.

It is necessary to find the minimal total route distance for The Frog to be moved to destination point.

Input

The first line contains five integers X_A, Y_A, X_B, Y_B $(-150 \leq X_A, Y_A, X_B, Y_B \leq 150)$ — Cartesian coordinates of two different points A and B, and K $(0 < K \leq 5)$ — the number of teleport modes. The second line contains K integers L_i $(0 < Li \leq 150)$ — the teleportation distance for each operation mode.

Output

In the first line output the minimal route distance.

In the second line output number M — the count of teleportations in the minimal route.

In the following M lines output 2 real numbers each with the accuracy up to 10^{-6} — coordinates (x_i, y_i), where The Frog will be found after i-th teleportation.

If there are several minimal routes, output any of them.

Examples

stdin	stdout

`1 1 4 5 3`
`2 3 5`
	`5`
	`2`
	`2.2000000000 2.6000000000`
	`4.0000000000 5.0000000000`

`0 0 18 0 2`
`10 5`
	`20`
	`3`
	`4 3`
	`14 3`
	`18 0`

Problem O. Scrooge.net

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

In the Scrooge Village they have finally got Internet access. Well, sort of... Direct cable line was routed across the village and into the city nearby. The villagers were proud of the fact that the Internet was finally available in the village and decided to get connected. The provider made a decision to install only one connection gate but let the villagers choose the point of connection. Naturally, the scroogers wanted to have all k houses connected to the network and save as much as possible on the wiring. It was decided that the communication wires are to be stretched from each house to the connection point no matter how complicated the mounting was.

Knowing each house location it is required to find a point on the cable line, which makes the total distance to all the houses minimal.

Input

First line contains two pairs of integers X_1, Y_1 ,X_2 ,Y_2 ($-100 \le X_1, Y_1, X_2, Y_2 \le 100$) — the Cartesian coordinates of two different points located on the cable line. Second line contains integer k ($1 \le k \le 10\,000$) — the number of houses. Following k lines contain two integers each X_i, Y_i ($-10\,000 \le X_i, Y_i \le 10\,000$) — i-th house coordinates.

Output

Write the total wire length needed and coordinates of the connection point. If there are several such points, output any. The answer is considered to be correct if absolute or relative error does not exceed 10^{-5}.

Examples

stdin	stdout

`2 0 6 0`
`1`
`2 3`
	`3.0000000000`
	`2.0000000000 0.0000000000`

Problem P. Superfactorial numeral system

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	256 megabytes

On the most perfect of all planets i1c5l various numeral systems are being used during programming contests. In the second division they use *a superfactorial numeral system.* In this system any positive integer is presented as a linear combination of numbers converse to factorials:

$$\frac{p}{q} = a_1 + \frac{a_2}{2!} + \frac{a_3}{3!} + \ldots + \frac{a_n}{n!}\,.$$

Here a_1 is non-negative integer, and integers a_k for $k \geq 2$ satisfy $0 \leq a_k < k$. The nonsignificant zeros in the tail of the superfactorial number designation $\frac{p}{q}$ are rejected. The task is to find out how the rational number $\frac{p}{q}$ is presented in the superfactorial numeral system.

Input

Single line contains two space-separated integers p and q ($1 \leq p \leq 10^6$, $1 \leq q \leq 10^6$).

Output

Single line should contain a sequence of space-separated integers $a_1, a_2, \ldots, a_n$, forming a number designation $\frac{p}{q}$ in the superfactorial numeral system. If several solution exist, output any of them.

Examples

stdin	stdout
`1 2`	`0 1`
`2 10`	`0 0 1 0 4`
`10 2`	`5`

Problem Q. Microcircuits

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

You probably know how microcircuits look like. First of all it is important to pay special attention to connections. Contacts on the circuit are connected by lines. If two lines do not intersect then there is no connection between respective contacts.

You are holding a circular circuit with n contacts around the borderline. You have to calculate the number of possibilities to put exactly k non-intersecting lines each connecting two contacts.

Input

Single line contains two integers n and k ($1 \le k \le n \le 40$).

Output

Output the required number.

Examples

stdin	stdout
4 2	2
4 3	0