Codeforces Round #140 (Div. 2)

A. Where do I Turn?

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

Trouble came from the overseas lands: a three-headed dragon Gorynych arrived. The dragon settled at point C and began to terrorize the residents of the surrounding villages.

A brave hero decided to put an end to the dragon. He moved from point A to fight with Gorynych. The hero rode from point A along a straight road and met point B on his way. The hero knows that in this land for every pair of roads it is true that they are either parallel to each other, or lie on a straight line, or are perpendicular to each other. He also knows well that points B and C are connected by a road. So the hero must either turn 90 degrees to the left or continue riding straight ahead or turn 90 degrees to the right. But he forgot where the point C is located.

Fortunately, a Brave Falcon flew right by. It can see all three points from the sky. The hero asked him what way to go to get to the dragon's lair.

If you have not got it, you are the falcon. Help the hero and tell him how to get him to point C: turn left, go straight or turn right.

At this moment the hero is believed to stand at point B, turning his back to point A.

Input

The first input line contains two space-separated integers x_a, y_a ($|x_a|, |y_a| \le 10^9$) — the coordinates of point A. The second line contains the coordinates of point B in the same form, the third line contains the coordinates of point C.

It is guaranteed that all points are pairwise different. It is also guaranteed that either point B lies on segment AC, or angle ABC is right.

Output

Print a single line. If a hero must turn left, print "LEFT" (without the quotes); If he must go straight ahead, print "TOWARDS" (without the quotes); if he should turn right, print "RIGHT" (without the quotes).

Examples

input
0 0 0 1 1 1
output
RIGHT

input
-1 -1 -3 -3 -4 -4
output
TOWARDS

input
-4 -6 -3 -7 -2 -6
output
LEFT

Note

The picture to the first sample:

The red color shows points A, B and C. The blue arrow shows the hero's direction. The green color shows the hero's trajectory.

The picture to the second sample:

B. Effective Approach

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

Once at a team training Vasya, Petya and Sasha got a problem on implementing linear search in an array.

According to the boys, linear search works as follows. The array elements in a pre-selected order are in turn compared with the number that you need to find. Once you find the array element that is equal to the required one, the search ends. The efficiency of the algorithm is the number of performed comparisons. The fewer comparisons the linear search has made, the more effective it is.

Vasya believes that a linear search would work better if it sequentially iterates through the elements, starting with the 1-st one (in this problem we consider the elements of the array indexed from 1 to n) and ending with the n-th one. And Petya says that Vasya is wrong: the search will need less comparisons if it sequentially iterates the elements starting from the n-th and ending with the 1-st one. Sasha argues that the two approaches are equivalent.

To finally begin the task, the teammates decided to settle the debate and compare the two approaches on an example. For this, they took an array that is a permutation of integers from 1 to n, and generated m queries of the form: find element with value b_i in the array. They want to calculate for both approaches how many comparisons in total the linear search will need to respond to all queries. If the first search needs fewer comparisons, then the winner of the dispute is Vasya. If the second one does, then the winner is Petya. If both approaches make the same number of comparisons, then Sasha's got the upper hand.

But the problem is, linear search is too slow. That's why the boys aren't going to find out who is right before the end of the training, unless you come in here. Help them to determine who will win the dispute.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of elements in the array. The second line contains n distinct space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the elements of array.

The third line contains integer m ($1 \le m \le 10^5$) — the number of queries. The last line contains m space-separated integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le n$) — the search queries. Note that the queries can repeat.

Output

Print two integers, showing how many comparisons Vasya's approach needs and how many comparisons Petya's approach needs. Separate the numbers by spaces.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
2 1 2 1 1
output
1 2

input
2 2 1 1 1
output
2 1

input
3 3 1 2 3 1 2 3
output
6 6

Note

In the first sample Vasya's approach will make one comparison (it starts with the 1-st element and immediately finds the required number), and Petya's approach makes two comparisons (first he compares with the 2-nd array element, doesn't find the search item

and compares with the 1-st element).

In the second sample, on the contrary, Vasya's approach will need two comparisons (first with 1-st element, and then with the 2-nd), and Petya's approach will find the required value in one comparison (the first comparison with the 2-nd element).

C. Flying Saucer Segments

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

An expedition group flew from planet ACM-1 to Earth in order to study the bipedal species (its representatives don't even have antennas on their heads!).

The flying saucer, on which the brave pioneers set off, consists of three sections. These sections are connected by a chain: the 1-st section is adjacent only to the 2-nd one, the 2-nd one — to the 1-st and the 3-rd ones, the 3-rd one — only to the 2-nd one. The transitions are possible only between the adjacent sections.

The spacecraft team consists of n aliens. Each of them is given a rank — an integer from 1 to n. The ranks of all astronauts are distinct. The rules established on the Saucer, state that an alien may move from section a to section b only if it is senior in rank to all aliens who are in the segments a and b (besides, the segments a and b are of course required to be adjacent). Any alien requires exactly 1 minute to make a move. Besides, safety regulations require that no more than one alien moved at the same minute along the ship.

Alien A is senior in rank to alien B, if the number indicating rank A, is more than the corresponding number for B.

At the moment the whole saucer team is in the 3-rd segment. They all need to move to the 1-st segment. One member of the crew, the alien with the identification number CFR-140, decided to calculate the minimum time (in minutes) they will need to perform this task.

Help CFR-140, figure out the minimum time (in minutes) that all the astronauts will need to move from the 3-rd segment to the 1-st one. Since this number can be rather large, count it modulo m.

Input

The first line contains two space-separated integers: n and m ($1 \le n, m \le 10^9$) — the number of aliens on the saucer and the number, modulo which you should print the answer, correspondingly.

Output

Print a single number — the answer to the problem modulo m.

Examples

input
1 10
output
2

input
3 8
output
2

Note

In the first sample the only crew member moves from segment 3 to segment 2, and then from segment 2 to segment 1 without any problems. Thus, the whole moving will take two minutes.

To briefly describe the movements in the second sample we will use value $R_i \to j$, which would correspond to an alien with rank i moving from the segment in which it is at the moment, to the segment number j. Using these values, we will describe the movements between the segments in the second sample: $R_3 \to 2$, $R_3 \to 1$, $R_2 \to 2$, $R_3 \to 2$, $R_3 \to 3$, $R_2 \to 1$, $R_3 \to 2$, $R_3 \to 1$, $R_1 \to 2$, $R_3 \to 2$, $R_3 \to 3$, $R_2 \to 2$, $R_3 \to 2$, $R_3 \to 1$, $R_2 \to 3$, $R_3 \to 2$, $R_3 \to 3$, $R_1 \to 1$, $R_3 \to 2$, $R_3 \to 1$, $R_2 \to 2$, $R_3 \to 2$, $R_3 \to 3$, $R_2 \to 1$, $R_3 \to 2$, $R_3 \to 1$; In total: the aliens need 26 moves. The remainder after dividing 26 by 8 equals 2, so the answer to this test is 2.

D. Naughty Stone Piles

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

There are n piles of stones of sizes $a_1, a_2, \ldots, a_n$ lying on the table in front of you.

During one move you can take one pile and add it to the other. As you add pile i to pile j, the size of pile j increases by the current size of pile i, and pile i stops existing. The cost of the adding operation equals the size of the added pile.

Your task is to determine the minimum cost at which you can gather all stones in one pile.

To add some challenge, the stone piles built up conspiracy and decided that each pile will let you add to it not more than k times (after that it can only be added to another pile).

Moreover, the piles decided to puzzle you completely and told you q variants (not necessarily distinct) of what k might equal.

Your task is to find the minimum cost for each of q variants.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of stone piles. The second line contains n space-separated integers: $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the initial sizes of the stone piles.

The third line contains integer q ($1 \le q \le 10^5$) — the number of queries. The last line contains q space-separated integers $k_1, k_2, \ldots, k_q$ ($1 \le k_i \le 10^5$) — the values of number k for distinct queries. Note that numbers k_i can repeat.

Output

Print q whitespace-separated integers — the answers to the queries in the order, in which the queries are given in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
5 2 3 4 1 1 2 2 3

output
9 8

Note

In the first sample one way to get the optimal answer goes like this: we add in turns the 4-th and the 5-th piles to the 2-nd one; then we add the 1-st pile to the 3-rd one; we add the 2-nd pile to the 3-rd one. The first two operations cost 1 each; the third one costs 2, the fourth one costs 5 (the size of the 2-nd pile after the first two operations is not 3, it already is 5).

In the second sample you can add the 2-nd pile to the 3-rd one (the operations costs 3); then the 1-st one to the 3-th one (the cost is 2); then the 5-th one to the 4-th one (the costs is 1); and at last, the 4-th one to the 3-rd one (the cost is 2).

E. Anniversary

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

There are less than 60 years left till the 900-th birthday anniversary of a famous Italian mathematician Leonardo Fibonacci. Of course, such important anniversary needs much preparations.

Dima is sure that it'll be great to learn to solve the following problem by the Big Day: You're given a set A, consisting of numbers l, $l + 1$, $l + 2$, ..., r; let's consider all its k-element subsets; for each such subset let's find the largest common divisor of Fibonacci numbers with indexes, determined by the subset elements. Among all found common divisors, Dima is interested in the largest one.

Dima asked to remind you that Fibonacci numbers are elements of a numeric sequence, where $F_1 = 1$, $F_2 = 1$, $F_n = F_{n-1} + F_{n-2}$ for $n \geq 3$.

Dima has more than half a century ahead to solve the given task, but you only have two hours. Count the residue from dividing the sought largest common divisor by m.

Input

The first line contains four space-separated integers m, l, r and k ($1 \leq m \leq 10^9$; $1 \leq l < r \leq 10^{12}$; $2 \leq k \leq r - l + 1$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use `cin`, `cout` streams or the `%I64d` specifier.

Output

Print a single integer — the residue from dividing the sought greatest common divisor by m.

Examples

input
10 1 8 2
output
3

input
10 1 8 3
output
1