Codeforces Round #183 (Div. 1)

A. Lucky Permutation Triple

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

Bike is interested in permutations. A permutation of length n is an integer sequence such that each integer from 0 to $(n - 1)$ appears exactly once in it. For example, $[0, 2, 1]$ is a permutation of length 3 while both $[0, 2, 2]$ and $[1, 2, 3]$ is not.

A permutation triple of permutations of length n (a, b, c) is called a Lucky Permutation Triple if and only if $\forall i (1 \le i \le n), a_i + b_i \equiv c_i \mod n$. The sign a_i denotes the i-th element of permutation a. The modular equality described above denotes that the remainders after dividing $a_i + b_i$ by n and dividing c_i by n are equal.

Now, he has an integer n and wants to find a Lucky Permutation Triple. Could you please help him?

Input

The first line contains a single integer n $(1 \le n \le 10^5)$.

Output

If no Lucky Permutation Triple of length n exists print `-1`.

Otherwise, you need to print three lines. Each line contains n space-seperated integers. The first line must contain permutation a, the second line — permutation b, the third — permutation c.

If there are multiple solutions, print any of them.

Sample test(s)

input
5

output
1 4 3 2 0 1 0 2 4 3 2 4 0 1 3

input
2

output
-1

Note

In Sample 1, the permutation triple $([1, 4, 3, 2, 0], [1, 0, 2, 4, 3], [2, 4, 0, 1, 3])$ is Lucky Permutation Triple, as following holds:

- $1 + 1 \equiv 2 \equiv 2 \mod 5$;
- $4 + 0 \equiv 4 \equiv 4 \mod 5$;
- $3 + 2 \equiv 0 \equiv 0 \mod 5$;
- $2 + 4 \equiv 6 \equiv 1 \mod 5$;
- $0 + 3 \equiv 3 \equiv 3 \mod 5$.

In Sample 2, you can easily notice that no lucky permutation triple exists.

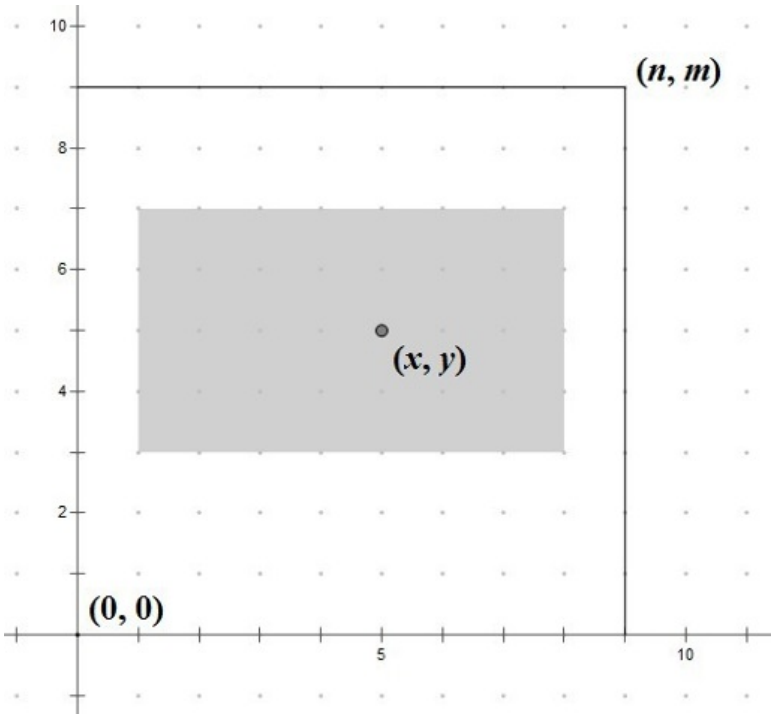
B. Rectangle Puzzle II

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

You are given a rectangle grid. That grid's size is $n \times m$. Let's denote the coordinate system on the grid. So, each point on the grid will have coordinates — a pair of integers (x, y) $(0 \le x \le n, 0 \le y \le m)$.

Your task is to find a maximum sub-rectangle on the grid (x_1, y_1, x_2, y_2) so that it contains the given point (x, y), and its length-width ratio is exactly (a, b). In other words the following conditions must hold: $0 \le x_1 \le x \le x_2 \le n$, $0 \le y_1 \le y \le y_2 \le m$, $\frac{x_2 - x_1}{y_2 - y_1} = \frac{a}{b}$.

The sides of this sub-rectangle should be parallel to the axes. And values x_1, y_1, x_2, y_2 should be integers.



If there are multiple solutions, find the rectangle which is closest to (x, y). Here "closest" means the Euclid distance between (x, y) and the center of the rectangle is as small as possible. If there are still multiple solutions, find the lexicographically minimum one. Here "lexicographically minimum" means that we should consider the sub-rectangle as sequence of integers (x_1, y_1, x_2, y_2), so we can choose the lexicographically minimum one.

Input

The first line contains six integers n, m, x, y, a, b $(1 \le n, m \le 10^9, 0 \le x \le n, 0 \le y \le m, 1 \le a \le n, 1 \le b \le m)$.

Output

Print four integers x_1, y_1, x_2, y_2, which represent the founded sub-rectangle whose left-bottom point is (x_1, y_1) and right-up point is (x_2, y_2).

Sample test(s)

input
9 9 5 5 2 1
output
1 3 9 7

input
100 100 52 50 46 56
output
17 8 86 92

C. Minimum Modular

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

You have been given n distinct integers $a_1, a_2, ..., a_n$. You can remove at most k of them. Find the minimum modular m $(m > 0)$, so that for every pair of the remaining integers (a_i, a_j), the following unequality holds: $a_i \not\equiv a_j \mod m$.

Input

The first line contains two integers n and k $(1 \leq n \leq 5000, 0 \leq k \leq 4)$, which we have mentioned above.

The second line contains n distinct integers $a_1, a_2, ..., a_n$ $(0 \leq a_i \leq 10^6)$.

Output

Print a single positive integer — the minimum m.

Sample test(s)

input
7 0 0 2 3 6 7 12 18
output
13

input
7 1 0 2 3 6 7 12 18
output
7

D. Rotatable Number

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

Bike is a smart boy who loves math very much. He invented a number called "Rotatable Number" inspired by 142857.

As you can see, 142857 is a magic number because any of its rotatings can be got by multiplying that number by $1, 2, ..., 6$ (numbers from one to number's length). Rotating a number means putting its last several digit into first. For example, by rotating number 12345 you can obtain any numbers: $12345, 51234, 45123, 34512, 23451$. It's worth mentioning that **leading-zeroes** are allowed. So both 4500123 and 0123450 can be obtained by rotating 0012345. You can see why 142857 satisfies the condition. All of the 6 equations are under base 10.

- $142857 \cdot 1 = 142857$;
- $142857 \cdot 2 = 285714$;
- $142857 \cdot 3 = 428571$;
- $142857 \cdot 4 = 571428$;
- $142857 \cdot 5 = 714285$;
- $142857 \cdot 6 = 857142$.

Now, Bike has a problem. He extends "Rotatable Number" under any base b. As is mentioned above, 142857 is a "Rotatable Number" under base 10. Another example is 0011 under base 2. All of the 4 equations are under base 2.

- $0011 \cdot 1 = 0011$;
- $0011 \cdot 10 = 0110$;
- $0011 \cdot 11 = 1001$;
- $0011 \cdot 100 = 1100$.

So, he wants to find the largest b $(1 < b < x)$ so that there is a **positive** "Rotatable Number" (leading-zeroes allowed) of length n under base b.

Note that any time you multiply a rotatable number by numbers from 1 to its length you should get a rotating of that number.

Input

The only line contains two space-separated integers n, x $(1 \le n \le 5 \cdot 10^6, 2 \le x \le 10^9)$.

Output

Print a single integer — the largest b you found. If no such b exists, print `-1` instead.

Sample test(s)

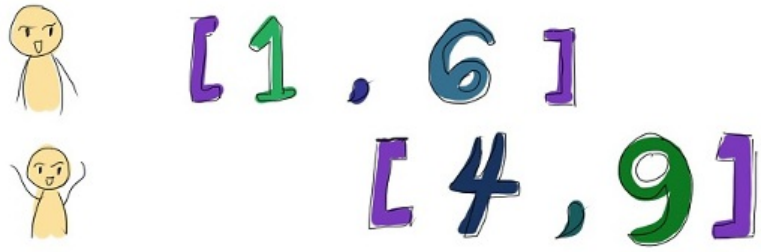
input
6 11
output
10

input
5 8
output
-1

E. Random Ranking

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

Imagine a real contest or exam of n participants. Every participant will get a particular score. We can predict the standings board more or less, if we do some statistics on their previous performance.



Let's say the score of the participants will be uniformly distributed in interval $[l_i, r_i]$ (the score can be a real number). Can you predict the standings board according to these data? In other words you should say for each participant the probability that he gets some fixed place in the scoreboard. **The participants are sorted by increasing of their scores in the scoreboard. So, the participant with the largest score gets the last place.**

Input

The first line contains integer n ($1 \le n \le 80$), showing how many participants we have. Each of the next n lines contains our predictions, the i-th line contains a pair of integers l_i, r_i ($0 \le l_i < r_i \le 10^9$) as the distributed interval for participant i.

Consider the participants numbered from 1 to n in some way.

Output

Output a distributed matrix a of order n. The element a_{ij} of the matrix is the probability that participant i has rank j.

Your answer will considered correct if it has at most 10^{-6} absolute or relative error.

Sample test(s)

input

```
2
1 6
4 9
```

output

```
0.9200000000 0.080
0.080 0.9200000000
```

input

```
8
0 2
1 3
2 4
3 5
4 6
5 7
6 8
7 9
```

output

```
0.875 0.125 0 0 0 0 0 0
0.125 0.750 0.125 0 0 0 0 0
0 0.125 0.750 0.125 0 0 0 0
0 0 0.125 0.750 0.125 0 0 0
0 0 0 0.125 0.750 0.125 0 0
0 0 0 0 0.125 0.750 0.125 0
0 0 0 0 0 0.125 0.750 0.125
0 0 0 0 0 0 0.125 0.875
```

Note

The score probability distribution is continuous, which means, there is no possibility for a draw.