Good Bye 2018

A. New Year and the Christmas Ornament

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob are decorating a Christmas Tree.

Alice wants only 3 types of ornaments to be used on the Christmas Tree: yellow, blue and red. They have y yellow ornaments, b blue ornaments and r red ornaments.

In Bob's opinion, a Christmas Tree will be beautiful if:

- the number of blue ornaments used is greater by **exactly** 1 than the number of yellow ornaments, and
- the number of red ornaments used is greater by **exactly** 1 than the number of blue ornaments.

That is, if they have 8 yellow ornaments, 13 blue ornaments and 9 red ornaments, we can choose 4 yellow, 5 blue and 6 red ornaments ($5=4+1$ and $6=5+1$).

Alice wants to choose as many ornaments as possible, but she also wants the Christmas Tree to be beautiful according to Bob's opinion.

In the example two paragraphs above, we would choose 7 yellow, 8 blue and 9 red ornaments. If we do it, we will use $7+8+9=24$ ornaments. That is the maximum number.

Since Alice and Bob are busy with preparing food to the New Year's Eve, they are asking you to find out the maximum number of ornaments that can be used in their **beautiful** Christmas Tree!

It is guaranteed that it is possible to choose at least 6 ($1+2+3=6$) ornaments.

Input

The only line contains three integers y, b, r ($1 \leq y \leq 100$, $2 \leq b \leq 100$, $3 \leq r \leq 100$) — the number of yellow, blue and red ornaments.

It is guaranteed that it is possible to choose at least 6 ($1+2+3=6$) ornaments.

Output

Print one number — the maximum number of ornaments that can be used.

Examples

input

```
8 13 9
```

output

```
24
```

input

```
13 3 6
```

output

```
9
```

Note

In the first example, the answer is $7+8+9=24$.

In the second example, the answer is $2+3+4=9$.

B. New Year and the Treasure Geolocation

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob is a pirate looking for the greatest treasure the world has ever seen. The treasure is located at the point T, which coordinates to be found out.

Bob travelled around the world and collected clues of the treasure location at n obelisks. These clues were in an ancient language, and he has only decrypted them at home. Since he does not know which clue belongs to which obelisk, finding the treasure might pose a challenge. Can you help him?

As everyone knows, the world is a two-dimensional plane. The i-th obelisk is at integer coordinates (x_i, y_i). The j-th clue consists of 2 integers (a_j, b_j) and belongs to the obelisk p_j, where p is some (unknown) permutation on n elements. It means that the treasure is located at $T = (x_{p_j} + a_j, y_{p_j} + b_j)$. This point T is the same for all clues.

In other words, each clue belongs to exactly one of the obelisks, and each obelisk has exactly one clue that belongs to it. A clue represents the vector from the obelisk to the treasure. The clues must be distributed among the obelisks in such a way that they all point to the same position of the treasure.

Your task is to find the coordinates of the treasure. If there are multiple solutions, you may print any of them.

Note that you don't need to find the permutation. Permutations are used only in order to explain the problem.

Input

The first line contains an integer n ($1 \le n \le 1000$) — the number of obelisks, that is also equal to the number of clues.

Each of the next n lines contains two integers x_i, y_i ($-10^6 \le x_i, y_i \le 10^6$) — the coordinates of the i-th obelisk. All coordinates are distinct, that is $x_i \neq x_j$ or $y_i \neq y_j$ will be satisfied for every (i, j) such that $i \neq j$.

Each of the next n lines contains two integers a_i, b_i ($-2 \cdot 10^6 \le a_i, b_i \le 2 \cdot 10^6$) — the direction of the i-th clue. All coordinates are distinct, that is $a_i \neq a_j$ or $b_i \neq b_j$ will be satisfied for every (i, j) such that $i \neq j$.

It is guaranteed that there exists a permutation p, such that for all i, j it holds $\left(x_{p_i} + a_i, y_{p_i} + b_i\right) = \left(x_{p_j} + a_j, y_{p_j} + b_j\right)$.

Output

Output a single line containing two integers T_x, T_y — the coordinates of the treasure.

If there are multiple answers, you may print any of them.

Examples

input

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

output

```
1 2
```

input

```
4
2 2
8 2
-7 0
-2 6
1 -14
16 -12
11 -18
7 -14
```

output

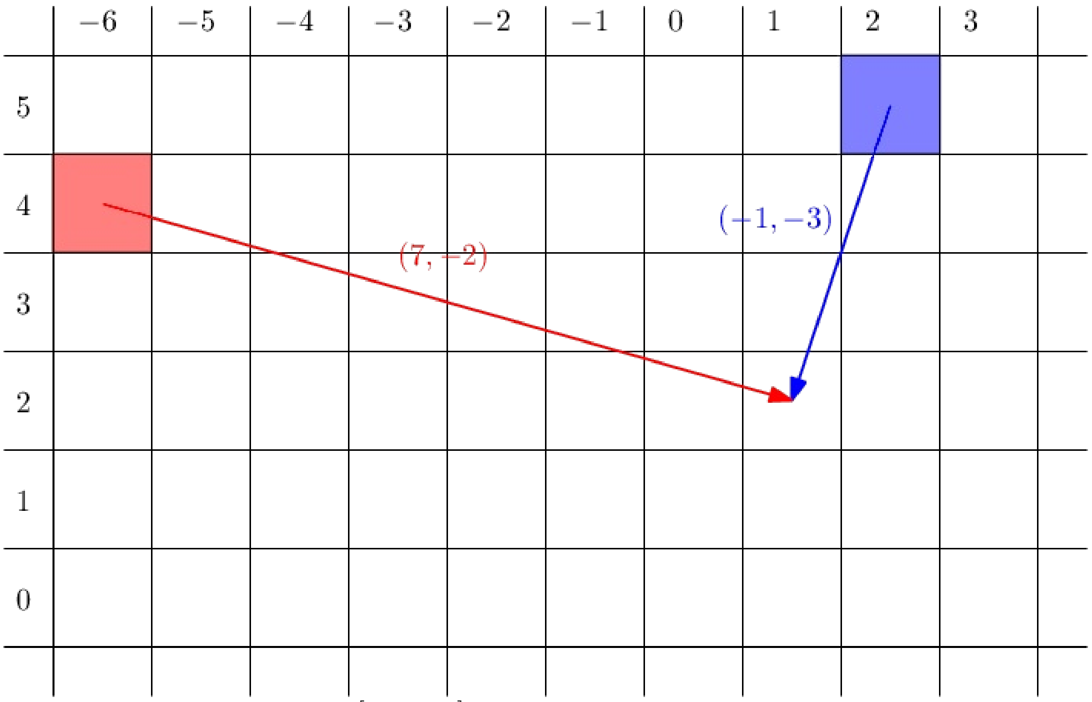
```
9 -12
```

Note

As $n = 2$, we can consider all permutations on two elements.

If $p = [1, 2]$, then the obelisk $(2, 5)$ holds the clue $(7, -2)$, which means that the treasure is hidden at $(9, 3)$. The second obelisk $(-6, 4)$ would give the clue $(-1, -3)$ and the treasure at $(-7, 1)$. However, both obelisks must give the same location, hence this is clearly not the correct permutation.

If the hidden permutation is $[2, 1]$, then the first clue belongs to the second obelisk and the second clue belongs to the first obelisk. Hence $(-6, 4) + (7, -2) = (2, 5) + (-1, -3) = (1, 2)$, so $T = (1, 2)$ is the location of the treasure.



In the second sample, the hidden permutation is $[2, 3, 4, 1]$.

C. New Year and the Sphere Transmission

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n people sitting in a circle, numbered from 1 to n in the order in which they are seated. That is, for all i from 1 to $n-1$, the people with id i and $i+1$ are adjacent. People with id n and 1 are adjacent as well.

The person with id 1 initially has a ball. He picks a positive integer k at most n, and passes the ball to his k-th neighbour in the direction of increasing ids, that person passes the ball to his k-th neighbour in the same direction, and so on until the person with the id 1 gets the ball back. When he gets it back, people do not pass the ball any more.

For instance, if $n = 6$ and $k = 4$, the ball is passed in order $[1, 5, 3, 1]$.

Consider the set of all people that touched the ball. The **fun value** of the game is the sum of the ids of people that touched it. In the above example, the *fun value* would be $1 + 5 + 3 = 9$.

Find and report the set of possible *fun values* for all choices of positive integer k. It can be shown that under the constraints of the problem, the ball always gets back to the 1-st player after finitely many steps, and there are no more than 10^5 possible *fun values* for given n.

Input

The only line consists of a single integer n ($2 \le n \le 10^9$) — the number of people playing with the ball.

Output

Suppose the set of all *fun values* is $f_1, f_2, \dots, f_m$.

Output a single line containing m space separated integers f_1 through f_m in **increasing** order.

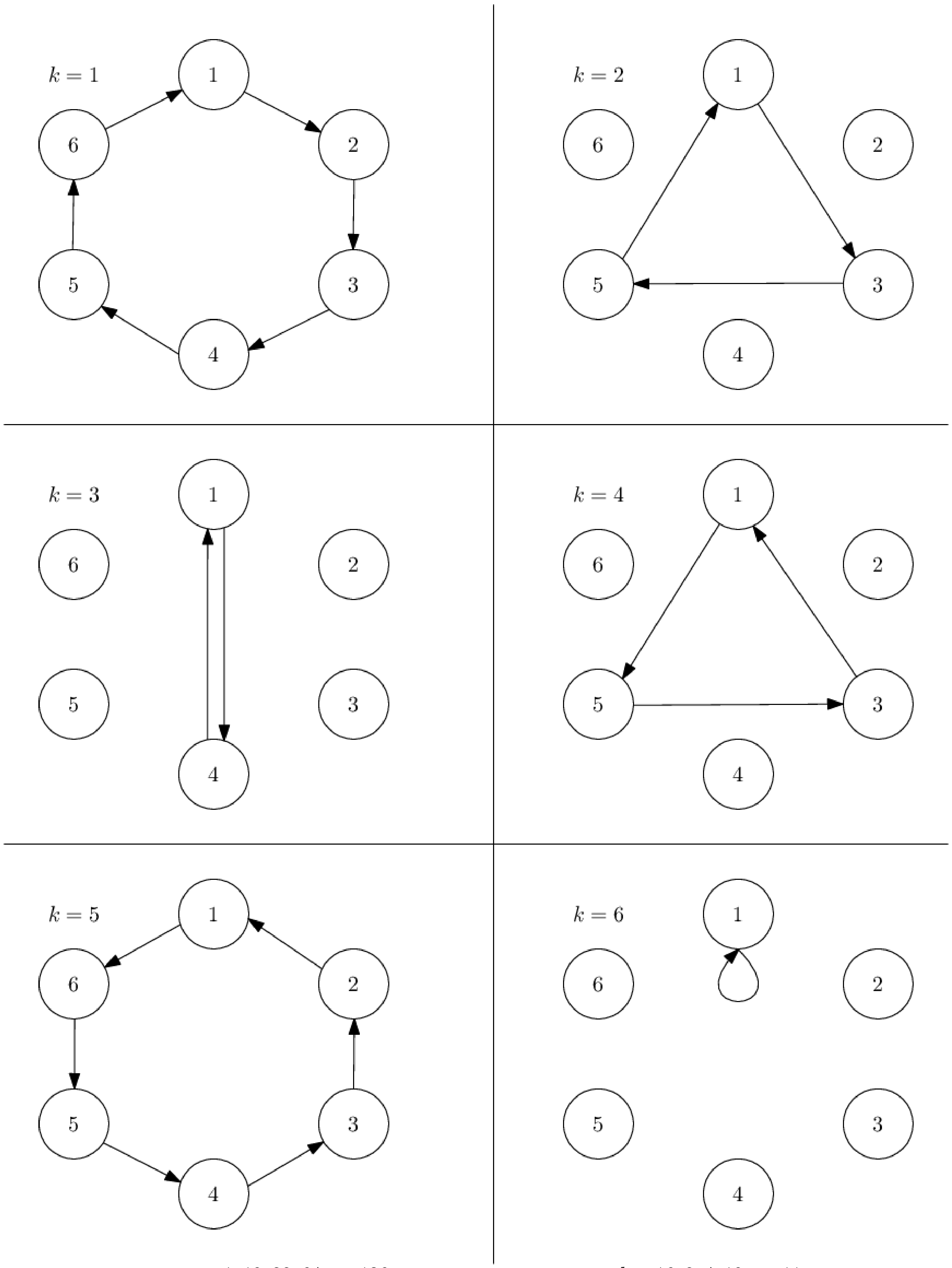
Examples

input
```
6
```
output
```
1 5 9 21
```

input
```
16
```
output
```
1 10 28 64 136
```

Note

In the first sample, we've already shown that picking $k = 4$ yields *fun value* 9, as does $k = 2$. Picking $k = 6$ results in *fun value* of 1. For $k = 3$ we get *fun value* 5 and with $k = 1$ or $k = 5$ we get 21.



In the second sample, the values 1, 10, 28, 64 and 136 are achieved for instance for $k = 16$, 8, 4, 10 and 11, respectively.

D. New Year and the Permutation Concatenation

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

Let n be an integer. Consider all permutations on integers 1 to n in lexicographic order, and concatenate them into one big sequence p. For example, if $n = 3$, then $p = [1, 2, 3, 1, 3, 2, 2, 1, 3, 2, 3, 1, 3, 1, 2, 3, 2, 1]$. The length of this sequence will be $n \cdot n!$.

Let $1 \leq i \leq j \leq n \cdot n!$ be a pair of indices. We call the sequence $(p_i, p_{i+1}, \dots, p_{j-1}, p_j)$ a **subarray** of p. Its **length** is defined as the number of its elements, i.e., $j - i + 1$. Its **sum** is the sum of all its elements, i.e., $\sum_{k=i}^{j} p_k$.

You are given n. Find the number of subarrays of p of length n having sum $\frac{n(n+1)}{2}$. Since this number may be large, output it modulo 998244353 (a prime number).

Input

The only line contains one integer n ($1 \le n \le 10^6$), as described in the problem statement.

Output

Output a single integer — the number of subarrays of length n having sum $\frac{n(n+1)}{2}$, modulo 998244353.

Examples

input
```
3
```
output
```
9
```

input
```
4
```
output
```
56
```

input
```
10
```
output
```
30052700
```

Note

In the first sample, there are 16 subarrays of length 3. In order of appearance, they are:

$[1, 2, 3]$, $[2, 3, 1]$, $[3, 1, 3]$, $[1, 3, 2]$, $[3, 2, 2]$, $[2, 2, 1]$, $[2, 1, 3]$, $[1, 3, 2]$, $[3, 2, 3]$, $[2, 3, 1]$, $[3, 1, 3]$, $[1, 3, 1]$, $[3, 1, 2]$, $[1, 2, 3]$, $[2, 3, 2]$, $[3, 2, 1]$.

Their sums are 6, 6, 7, 6, 7, 5, 6, 6, 8, 6, 7, 5, 6, 6, 7, 6. As $\frac{n(n+1)}{2} = 6$, the answer is 9.

E. New Year and the Acquaintance Estimation

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

Bob is an active user of the social network Faithbug. On this network, people are able to engage in a mutual friendship. That is, if a is a friend of b, then b is also a friend of a. Each user thus has a non-negative amount of friends.

This morning, somebody anonymously sent Bob the following link: graph realization problem and Bob wants to know who that was. In order to do that, he first needs to know how the social network looks like. He investigated the profile of every other person on the network and noted down the number of his friends. However, he neglected to note down the number of his friends. Help him find out how many friends he has. Since there may be many possible answers, print **all** of them.

Input

The first line contains one integer n ($1 \le n \le 5 \cdot 10^5$), the number of people on the network **excluding** Bob.

The second line contains n numbers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le n$), with a_i being the number of people that person i is a friend of.

Output

Print all possible values of a_{n+1} — the amount of people that Bob can be friend of, **in increasing order**.

If no solution exists, output -1.

Examples

input
```
3
3 3 3
```
output
```
3
```

input
```
4
```

```
1 1 1 1
```

output

```
0 2 4
```

input

```
2
0 2
```

output

```
-1
```

input

```
35
21 26 18 4 28 2 15 13 16 25 6 32 11 5 31 17 9 3 24 33 14 27 29 1 20 4 12 7 10 30 34 8 19 23 22
```

output

```
13 15 17 19 21
```

Note

In the first test case, the only solution is that everyone is friends with everyone. That is why Bob should have 3 friends.

In the second test case, there are three possible solutions (apart from symmetries):

- a is friend of b, c is friend of d, and Bob has no friends, or
- a is a friend of b and both c and d are friends with Bob, or
- Bob is friends of everyone.

The third case is impossible to solve, as the second person needs to be a friend with everybody, but the first one is a complete stranger.

F. New Year and the Mallard Expedition

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob is a duck. He wants to get to Alice's nest, so that those two can duck!

Duck is the ultimate animal! (Image courtesy of See Bang)

The journey can be represented as a straight line, consisting of n segments. Bob is located to the left of the first segment, while Alice's nest is on the right of the last segment. Each segment has a length in meters, and also terrain type: grass, water or lava.

Bob has three movement types: swimming, walking and flying. He can switch between them or change his direction at any point in time (even when he is located at a non-integer coordinate), and doing so doesn't require any extra time. Bob can swim only on the water, walk only on the grass and fly over **any** terrain. Flying one meter takes 1 second, swimming one meter takes 3 seconds, and finally walking one meter takes 5 seconds.

Bob has a finite amount of energy, called stamina. Swimming and walking is relaxing for him, so he gains 1 stamina for every meter he walks or swims. On the other hand, flying is quite tiring, and he spends 1 stamina for every meter flown. Staying in place does not influence his stamina at all. Of course, his stamina can never become negative. Initially, his stamina is zero.

What is the shortest possible time in which he can reach Alice's nest?

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of segments of terrain.

The second line contains n integers $l_1, l_2, \ldots, l_n$ ($1 \leq l_i \leq 10^{12}$). The l_i represents the length of the i-th terrain segment in meters.

The third line contains a string s consisting of n characters "G", "W", "L", representing **G**rass, **W**ater and **L**ava, respectively.

It is guaranteed that the first segment is not Lava.

Output

Output a single integer t — the minimum time Bob needs to reach Alice.

Examples

input

```
1
10
G
```

output

```
30
```

input

```
2
10 10
WL
```

output

```
40
```

input

```
2
1 2
WL
```

output

```
8
```

input

```
3
10 10 10
GLW
```

output

```
80
```

Note

In the first sample, Bob first walks 5 meters in 25 seconds. Then he flies the remaining 5 meters in 5 seconds.

In the second sample, Bob first swims 10 meters in 30 seconds. Then he flies over the patch of lava for 10 seconds.

In the third sample, the water pond is much smaller. Bob first swims over the water pond, taking him 3 seconds. However, he cannot fly over the lava just yet, as he only has one stamina while he needs two. So he swims back for half a meter, and then half a meter forward, taking him 3 seconds in total. Now he has 2 stamina, so he can spend 2 seconds flying over the lava.

In the fourth sample, he walks for 50 seconds, flies for 10 seconds, swims for 15 seconds, and finally flies for 5 seconds.

G. New Year and the Factorisation Collaboration

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Integer factorisation is hard. The RSA Factoring Challenge offered $\$100\,000$ for factoring RSA-1024, a 1024-bit long product of two prime numbers. To this date, nobody was able to claim the prize. We want you to factorise a 1024-bit number.

Since your programming language of choice might not offer facilities for handling large integers, we will provide you with a very simple calculator.

To use this calculator, you can print queries on the standard output and retrieve the results from the standard input. The operations are as follows:

- `+ x y` where x and y are integers between 0 and $n-1$. Returns $(x+y) \bmod n$.
- `- x y` where x and y are integers between 0 and $n-1$. Returns $(x-y) \bmod n$.
- `* x y` where x and y are integers between 0 and $n-1$. Returns $(x \cdot y) \bmod n$.
- `/ x y` where x and y are integers between 0 and $n-1$ and y is coprime with n. Returns $(x \cdot y^{-1}) \bmod n$ where y^{-1} is

multiplicative inverse of y modulo n. If y is not coprime with n, then -1 is returned instead.
- `sqrt x` where x is integer between 0 and $n-1$ coprime with n. Returns y such that $y^2 \bmod n = x$. If there are multiple such integers, only one of them is returned. If there are none, -1 is returned instead.
- `^ x y` where x and y are integers between 0 and $n-1$. Returns $x^y \bmod n$.

Find the factorisation of n that is a product of between 2 and 10 **distinct** prime numbers, all of form $4x+3$ for some integer x.

Because of technical issues, we restrict number of requests to 100.

Input

The only line contains a single integer n ($21 \le n \le 2^{1024}$). It is guaranteed that n is a product of between 2 and 10 distinct prime numbers, all of form $4x+3$ for some integer x.

Output

You can print as many queries as you wish, adhering to the time limit (see the Interaction section for more details).

When you think you know the answer, output a single line of form `! k p_1 p_2 ... p_k`, where k is the number of prime factors of n, and p_i are the distinct prime factors. You may print the factors in any order.

Hacks input

For hacks, use the following format:.

The first should contain k ($2 \le k \le 10$) — the number of prime factors of n.

The second should contain k space separated integers $p_1, p_2, \ldots, p_k$ ($21 \le n \le 2^{1024}$) — the prime factors of n. All prime factors have to be of form $4x+3$ for some integer x. They all have to be distinct.

Interaction

After printing a query do not forget to output end of line and flush the output. Otherwise you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

The number of queries is not limited. However, your program must (as always) fit in the time limit. The run time of the interactor is also counted towards the time limit. The maximum runtime of each query is given below.

- `+ x y` — up to 1 ms.
- `- x y` — up to 1 ms.
- `* x y` — up to 1 ms.
- `/ x y` — up to 350 ms.
- `sqrt x` — up to 80 ms.
- `^ x y` — up to 350 ms.

Note that the sample input contains extra empty lines so that it easier to read. The real input will not contain any empty lines and you do not need to output extra empty lines.

Example

input

```
21

7

17

15

17

11

-1

15
```

output

```
+ 12 16

- 6 10

* 8 15

/ 5 4

sqrt 16
sqrt 5

^ 6 12

! 2 3 7
```

Note

We start by reading the first line containing the integer $n = 21$. Then, we ask for:

1. $(12 + 16) \bmod 21 = 28 \bmod 21 = 7$.
2. $(6 - 10) \bmod 21 = -4 \bmod 21 = 17$.
3. $(8 \cdot 15) \bmod 21 = 120 \bmod 21 = 15$.
4. $(5 \cdot 4^{-1}) \bmod 21 = (5 \cdot 16) \bmod 21 = 80 \bmod 21 = 17$.
5. Square root of 16. The answer is 11, as $(11 \cdot 11) \bmod 21 = 121 \bmod 21 = 16$. Note that the answer may as well be 10.
6. Square root of 5. There is no x such that $x^2 \bmod 21 = 5$, so the output is -1.
7. $(6^{12}) \bmod 21 = 2176782336 \bmod 21 = 15$.

We conclude that our calculator is working, stop fooling around and realise that $21 = 3 \cdot 7$.

H. New Year and the Tricolore Recreation

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob play a game on a grid with n rows and infinitely many columns. In each row, there are three tokens, blue, white and red one. **Before** the game starts and **after every move**, the following two conditions must hold:

- Any two tokens are not in the same cell.
- In each row, the blue token is to the left of the white token, and the red token is to the right of the white token.

First, they pick a positive integer f, whose value is valid for the whole game. Second, the starting player is chosen and makes his or her first turn. Then players take alternating turns. The player who is unable to make a move loses.

During a move, a player first selects an integer k that is either a prime number or a product of two (not necessarily distinct) primes. The smallest possible values of k are thus $2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 19, \dots$. Furthermore, k must not be equal to the previously picked integer f. Each turn, a move is performed in exactly one of the rows.

If it is Alice's turn, she chooses a single blue token and moves it k cells to the right. Alternatively, she may move both the blue and the white token in the same row by the same amount k to the right.

On the other hand, Bob selects a single red token and moves it k cells to the left. Similarly, he may also move the white and the red token in the corresponding row by k to the left.

Note that Alice may never move a red token, while Bob may never move a blue one. Remember that after a move, the two conditions on relative positions of the tokens must still hold.

Both players play optimally. Given the initial state of the board, determine who wins for two games: if Alice starts and if Bob starts.

Input

The first line contains a two integers n and f ($1 \le n \le 10^5$, $2 \le f \le 2 \cdot 10^5$) — the number of rows and the forbidden move, respectively.

Each of the next n lines contains three integers b_i, w_i, r_i ($-10^5 \le b_i < w_i < r_i \le 10^5$) — the number of column in which the blue, white and red token lies in the i-th row, respectively.

Output

Output two lines.

The first line should contain the name of the winner when Alice starts, and the second line should contain the name of the winner when Bob starts.

Examples

input

```
1 6
0 3 9
```

output

```
Alice
Bob
```

input

```
1 2
0 3 9
```

output

```
Alice
Bob
```

input

```
10 133
-248 -193 -187
97 101 202
-72 67 91
23 89 215
-129 -108 232
-223 -59 236
-99 86 242
-137 -109 -45
-105 173 246
-44 228 243
```

output

```
Bob
Alice
```

Note

The first example is as follows:

When Alice starts, she can win by moving the blue and white token to right by 2 cells, getting into position $2\ 5\ 9$. Regardless of what Bob does, Alice will have one more move and then the game is over. For instance, he can move both the red and white token by 2 cells to the left, reaching state $2\ 3\ 7$. Alice can then move blue and white token by 2 to move into $4\ 5\ 7$, where no more moves are possible.

If Bob starts, he gains enough advantage to win. For instance, he may move the red token by 3 to the left, getting into position $0\ 3\ 6$. Alice can, for example, move the blue token by 2, which is countered by Bob by moving the red token by 2. The game ends in position $2\ 3\ 4$.

In the second example, it is forbidden to move by 2, but this doesn't stop Alice from winning! She can move the blue and white token by 4, getting into position $4\ 7\ 9$. Now Bob has no move, since moving by 2 is forbidden.