icpc International Collegiate Programming Contest

The 2024 ICPC Rocky Mountain Regional Contest

Problem Set

icpc.foundation

icpc Rocky Mountain

JETBRAINS

icpc global sponsor
programming tools

upsilon pi epsilon
honor society

NOVEMBER 9, 2024

Contest Problems

The 2024 ICPC Rocky Mountain Regional Contest

icpc.foundation
JETBRAINS
icpc global sponsor programming tools
upsilon pi epsilon honor society

This contest contains 13 problems over 30 pages. Good luck.

For problems that state "*Your answer should have a relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

For problems that state "*Your answer should have an absolute error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \le a < m$

  and

- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \le \min(n, \ell)$ such that $s_i = t_i$ for all $1 \le i < k$ and $s_k < t_k$

  or

- $s_i = t_i$ for all $1 \le i \le \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

This page is intentionally left blank.

Problem A
Airfare Grants
Time limit: 1 second

AI-generated image of Cupcake City.

Congratulations! Your team has advanced to the next round of the International Cupcake Production Competition (ICPC)! To participate in the competition, you are going to fly to Cupcake City from your hometown. The competition organizers have just shared good news: generous sponsors will support you with your flight ticket!

Here's how it works. You have already obtained a list of available flights. Among those options, you first report the price of any one potential flight. This determines the *reimbursement limit* of your trip; sponsors would pay you *up to half that price*. Next, you actually purchase your flight ticket. This flight may or may not be the same as the flight you report for the reimbursement limit. Finally, you show the receipt of your purchase, and sponsors will reimburse you the minimum of your actual cost and the reimbursement limit.

Now, you want to figure out your minimum possible *net* cost—the price you pay, minus the amount you get reimbursed. Naturally, you will report the price of the most expensive flight ticket and buy the cheapest flight ticket.

Input

The first line of input is an integer, N ($1 \leq N \leq 50$), the number of available flights.

Each of the next N lines contains an integer P ($10 \leq P \leq 10^5$), which is the price of a flight ticket from your hometown to Cupcake City, in dollars. It is guaranteed that P is a multiple of 10.

Output

Output a single integer, the minimum amount, in dollars, that you have to pay for a flight ticket, excluding the amount of travel support from sponsors.

Sample Input 1	Sample Output 1
2 150 250	25

Sample Input 2	Sample Output 2
5 200 300 150 400 500	0

Problem B
Always Know Where Your Towel Is
Time limit: 4 seconds

Arthur wants to go swimming. Another chance to use his favourite towel! He is concerned that if he leaves it out at the pool then someone might take it. So he keeps it in a locker.

Accessing the lockers is done in a peculiar way. One has to use an app on their phone to unlock the lockers. The app displays N distinct positive integers whose sum is at most $2^N - 2$. The lockers are also numbered from 1 to $2^N - 2$. To open locker S, one simply pushes a subset of buttons on the app (a "combination") whose sum is exactly S. Pushing a button multiple times has the same effect as pushing it once, so there is no point to pushing a button more than once.

Arthur makes some observations. Oddly, it might be impossible to open some lockers. Also, since there are $2^N - 1$ nonempty subsets of buttons and each such subset has its values summing to an integer from 1 to $2^N - 2$, then there must be at least one locker that can be opened with more than one combination.

Arthur does not want to forget where he puts his towel when he goes swimming. He believes it will be easier to remember which locker he uses if it has more than one combination that can open it. Help Arthur find such a locker!

Input

The first line of input contains a single integer N ($3 \leq N \leq 42$). Then next and final line contains N positive integers $a_1, \ldots, a_N$. These integers are distinct and satisfy $\sum_{i=1}^{N} a_i \leq 2^N - 2$.

Output

Output a single integer S ($1 \leq S \leq 2^N - 2$) such that locker S can be opened by more than one combination. If there are multiple possible solutions, any will suffice.

Sample Input 1
```
5
2 5 8 4 7
```
Sample Output 1
```
9
```

Sample Input 2
```
3
1 2 3
```
Sample Output 2
```
3
```

Sample Input 3
```
7
19 24 11 23 9 21 18
```
Sample Output 3
```
41
```

Problem C
Big And
Time limit: 3 seconds

You are building a boolean circuit and need to compute the logical `AND` of a collection of *source signals*. The problem is that you only have access to a collection of identical `AND` gates that take two boolean inputs (i.e. `True` or `False`) and outputs a single boolean value that is `True` if and only if both inputs were `True`.

Recalling your digital design course, you remember you can connect some of these `AND` gates together (by connecting the outputs of some to the inputs of others) to build a circuit that has one input for each source signal you need to consider and a single output that is `True` if and only if every source signal is `True`. More precisely, you should build a circuit satisfying the following properties.

- There are precisely $N - 1$ `AND` gates in the circuit where N is the number of source signals.
- Every source signal will be connected to the input of precisely one `AND` gate.
- Every input to an `AND` gate in the circuit is connected to precisely one signal, which will either be one of the source signals or will be the output signal of another `AND` gate.
- There will be no "cycle" of signals: the signal output from one `AND` gate will never reach one of its own input wires.

Finally, the output of your circuit will be connected to an LED which will be used to indicate if this signal is `True` or `False`. So the LED will illuminate if and only if all source signals are `True`.

You want to do this in a way that minimizes the worst-case delay for the output signal to change if the source of one of the input signals changes. You know the following:

- If the source signal i changes its value then it takes exactly n_i nanoseconds for this change to reach your circuit.
- For any `AND` gate, after an input signal changes it takes exactly D nanoseconds for the output signal to change (if the new inputs would cause the output to change).
- It takes exactly L nanoseconds for the LED to change when the signal it receives changes.
- Since the components of your circuit will be placed so close together, the time it takes for a signal to propagate from the output of one of your `AND` gates to another component of the circuit is negligible and will be regarded as 0 nanoseconds.

Help design a circuit to minimize the maximum time between when one of the sources signal changes to when the LED changes (if indeed the new input would cause the LED to change).

Input

The first line of input contains three integers N ($2 \leq N \leq 4 \cdot 10^5$), D and L ($1 \leq D, L \leq 10^9$) where N is the number of source signals and D, L are as in the problem description. The second line contains N

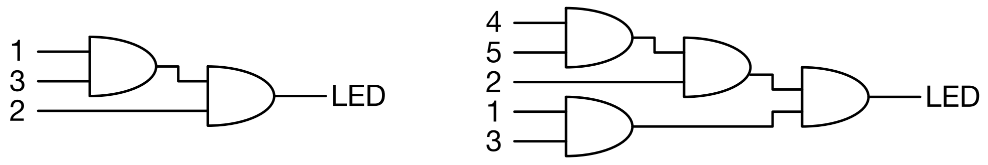

Figure C.1: Illustration of an optimal circuit for each of the sample inputs. The numbers on the left indicate the index of the source signal.

integers $n_1, n_2, \ldots n_N$ $(1 \leq n_i \leq 10^9)$ where n_i indicates the number of nanoseconds it takes for a change source i's signal to reach your circuit.

Output

Output the smallest time T such that it is possible to build a circuit ensuring it takes at most T nanoseconds for the LED to change if one of the source's signal changes in a way that would cause the LED to change.

Sample Input 1	Sample Output 1
3 3 5 3 8 1	16

Sample Input 2	Sample Output 2
5 5 10 5 8 3 3 3	28

Problem D
Emoticons
Time limit: 3 seconds

Emma is a college student who regularly posts messages on her favorite social media, which supports several emoticons—a pictorial representation of a facial expression using characters. One day, she noticed that some emoticons in her message were automatically converted to the corresponding emojis by the platform. For example, if she sends a message “Hello ;-)” to the platform, the actual post will be “Hello 😉”. If two or more emoticons overlap in a message, the system always converts the leftmost emoticon first.

Here is the complete list of the supported emoticons on that social media.

Type	Emoticon	Emoji	Meaning
Western	:)	🙂	Smiley
Western	:-)	🙂	Smiley
Western	:-(	🙁	Frown
Western	;-)	😉	Wink
Western	xD	😆	Laughing
Eastern	^_^	🙂	Smiley
Eastern	-_-	😑	Expressionless
Eastern	^o^	😀	Screwed
Eastern	^^;	😅	Sweating
Eastern	(..)	😔	Looking down

She always uses the following “visible” characters and the space character “ ”.

- Digits: 0123456789
- Uppercase letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Lowercase letters: abcdefghijklmnopqrstuvwxyz
- Symbols: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

Today, just before posting, she accidentally replaced all occurrences of one character in her message with another character. Such characters include space, and she might have replaced a character with the same character, causing no effect. Now, she wants to know the message length after the platform converts emoticons into emojis, where she counts each emoji as one character.

Can you help her find the minimum and maximum possible message lengths?

Input

The input contains Emma's original text (before she accidentally replaced characters) on one line. The text contains at most 100 characters, consisting of the visible ASCII characters defined above and the space character. Her text is not empty, and its first and last characters are visible characters.

Output

Output two integers, the minimum and the maximum possible message lengths after emoticons in her message are converted into emojis.

Sample Input 1

```
Hello ;-) world ^^.
```

Sample Output 1

```
15 19
```

Sample Input 2

```
xDEoE_E
```

Sample Output 2

```
4 7
```

Problem E
Enchanted Maze
Time limit: 3 seconds

Mila and her twin brother Liam have been trapped by an evil witch. The witch put the twins at different locations in a maze, and they must find their way out of the maze before she releases her pet dragon. The twins have asked you to help them escape the maze in the least amount of time.

To make things even more difficult, the witch has placed a curse on the twins. Although the twins can attempt to move in any of the four compass directions (North, South, East, West), they must attempt to move in the same direction at the same time. For example, if Mila attempts to walk North for a step, Liam must also attempt to walk North at the same time. Each attempted move by the twins takes one second to complete, regardless of whether the move was successful or not.

The maze is represented by a rectangular grid, where each location can be one of the following types:

- Start (`S`): the starting positions of the twins. There are exactly two starting locations, one for each twin.
- Exit (`E`): the twins escape if they are both at exit locations at the same time. There are exactly two exit locations.
- Open (`.`): the twins may freely move in and out of this location.
- Fixed Obstacle (`#`): the twins may not move into this location.
- Bottomless pit (`*`): a twin moving into a pit is lost forever.
- Raised obstacle (`A, B, C, D`): these obstacles can be lowered by stepping on the corresponding switch.
- Lowered obstacle (`a, b, c, d`): these locations are considered open, but they can be raised by stepping on the corresponding switch.
- Switch (`1, 2, 3, 4`): switches that control the obstacles. Switch 1 controls obstacles labelled `A` and `a`, switch 2 controls obstacles labelled `B` and `b`, switch 3 controls obstacles labelled `C` and `c`, and switch 4 controls obstacles labelled `D` and `d`. There is at most one switch of each label, but each switch may control more than one obstacle.

The twins may not be in the same location at the same time. If a twin attempts to move into a fixed or raised obstacle, the twin will instead remain stationary. A twin that moves outside of the boundary of the maze is lost forever. They are also allowed to enter and leave the starting and exit locations throughout their journey.

A switch is triggered when a twin moves onto a switch. When a switch is triggered, the corresponding obstacles are raised (if currently lowered) or lowered (if currently raised). The switch is not triggered if a twin is already on the switch and remains stationary after an attempted move.

If a twin attempts to move into a raised obstacle at the same time the other twin steps on the corresponding switch, the obstacle is still considered raised and is only lowered at the end of the move. A twin may not move onto a switch if the other twin is moving onto or remaining stationary at a corresponding lowered obstacle at the same time.

Input

The first line of input contains two integers, $2 \leq R \leq 10$ and $2 \leq C \leq 10$, specifying the number of rows and columns in the maze. The next R lines each contains C characters. All possible characters are specified above. It is always possible for the twins to escape the maze.

Output

Output the least number of seconds required for the twins to escape.

Sample Input 1

```
6 4
...E
S***
#***
S***
...E
.***
```

Sample Output 1

```
6
```

Sample Input 2

```
5 2
cE
E.
..
3.
SS
```

Sample Output 2

```
4
```

Sample Input 3

```
3 10
..........
.SS....EE.
..........
```

Sample Output 3

```
6
```

Problem F
Garage Door Code
Time limit: 4 seconds

There have been an increasing number of laptop thefts in your area lately. After yours was recently stolen, too, you decided that you had enough and are going to do something about it! After asking around, you heard some rumors about where the stolen laptops have been hidden and you want to get your laptop back!

The rumors led you to a particular garage and you have decided to keep an eye on it to try to figure out how to get in. Over the course of the afternoon, you have seen a handful of people come, enter the garage, and then leave.

There's a security keypad on the door. As you have watched carefully, you have observed different people putting in the code, but they enter different numbers of digits. You figure out that some, maybe all, of them are pretending to intersperse additional key presses just to confuse anyone who might be watching them. You look up the keypad manufacturer's webpage and find that the keypad was designed to use codes with a predetermined length of K digits. Given the set of observations you've made of different individuals "entering" elongated variations of the code (each does contain the correct code), can you determine what the correct code might be? It is guaranteed that there will always be at least one correct code that matches all observations. There may be multiple codes that match; you must find all matching codes.

Input

The first line of input contains two integers K ($3 \leq K \leq 6$), the number of digits specified by the manufacturer for a correct code, and N ($2 \leq N \leq 20$), the number of elongated entry code sequences that you observed.

Each of the next N lines contains a string of at least K and at most 12 digits (0-9). These are the elongated entry code sequences that you have observed.

Output

Determine all K-digit codes that match the full set of elongated observations.

On the first line, output M, the number of codes that match the set of elongated observations. Then output M lines, each containing one of these matching codes, having sorted them in increasing lexicographic order.

Sample Input 1

```
4 3
21302749
5230248
21076724
```

Sample Output 1

```
1
2024
```

Sample Input 2

```
5 7
0812734225
162334559
7512334755
1002394561
9321423495
81237425
47126345119
```

Sample Output 2

```
1
12345
```

Sample Input 3

```
3 3
1234
012345
11223344
```

Sample Output 3

```
4
123
124
134
234
```

Problem G
Palindromic Word Search
Time limit: 6 seconds

Given a rectangular grid of uppercase letters, find a rectangular region of the grid of maximum possible area such that there is a horizontal palindrome spanning some row of the rectangular region and a vertical palindrome spanning some column of the rectangular region. Recall a palindrome is a string that equals its own reversal.

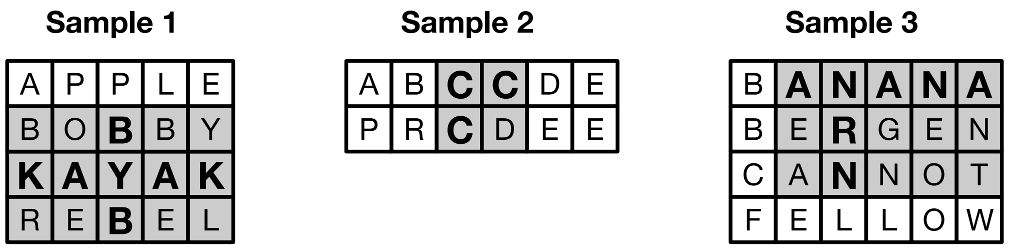

Figure G.1: Illustration of optimal solutions to the sample inputs. In the shaded subregions, a palindrome spanning an entire row of the region and a palindrome spanning an entire column of the region are highlighted.

Input

The first line contains two integers R and C ($1 \leq R, C \leq 500$). Then next R lines describe the grid, each containing exactly C uppercase letters.

Output

Output a single integer A indicating the largest area of a rectangular region of the grid such that there is a horizontal palindrome spanning some row and a vertical palindrome spanning some column of the rectangular region.

Sample Input 1	Sample Output 1
4 5 APPLE BOBBY KAYAK REBEL	15

Sample Input 2

```
2 6
ABCCDE
PRCDEE
```

Sample Output 2

```
4
```

Sample Input 3

```
4 6
BANANA
BERGEN
CANNOT
FELLOW
```

Sample Output 3

```
15
```

Problem H
Pillow Stacking
Time limit: 3 seconds

Eleanor has recently lost her favorite pillow. Being very particular about her pillows, she categorizes pillows by their softness. The pillow she lost had softness C, and she will be unable to sleep without this exact softness. While she has access to some other types of pillows, there may not be any of her desired softness. As a work-around, she is willing to sleep with multiple pillows in a stack. When pillows with softness $C_1, C_2, \ldots, C_k$ are stacked together in that order, the resulting softness is equal to

$$C_1 + \left\lceil \frac{C_2}{2} \right\rceil + \left\lceil \frac{C_3}{4} \right\rceil + \ldots + \left\lceil \frac{C_k}{2^{k-1}} \right\rceil .$$

In other words, the ith pillow in the stack contributes $2^{-(i-1)}$ of its softness, rounded up. Note that $\lceil f \rceil$ is defined as the smallest integer x with $x \geq f$.

Eleanor has a lot of money and is willing to use an unlimited amount of each pillow type she has access to. Can you help her determine if there is a way to stack pillows to reach her desired softness?

Input

The first line of input contains two integers N and C ($1 \leq N \leq 10^3$ and $1 \leq C \leq 10^4$). N indicates the number of types of pillows Eleanor has access to, and C indicates the desired softness level. The next line contains N integers $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq 10^{18}$), indicating the softness of pillows that Eleanor has access to.

Output

If it is possible for Eleanor to stack pillows to reach softness C, output `YES`. Otherwise, output `NO`.

Sample Input 1	Sample Output 1
1 1000 1	YES

Sample Input 2	Sample Output 2
1 5 4	NO

Sample Input 3	Sample Output 3
2 100 51 98	YES

This page is intentionally left blank.

Problem I
Rectangle Tiling
Time limit: 1 second

Consider a rectangle with integer side lengths. A square tiling of the rectangle is a covering of the entire region using non-overlapping squares whose sides are parallel with those of the rectangle. In a square tiling, no square may overhang (extend beyond the rectangle's boundary).

You have a collection of squares with side lengths being powers of 2. Find a square tiling of the rectangle using the fewest squares possible, or, indicate that it cannot be done.

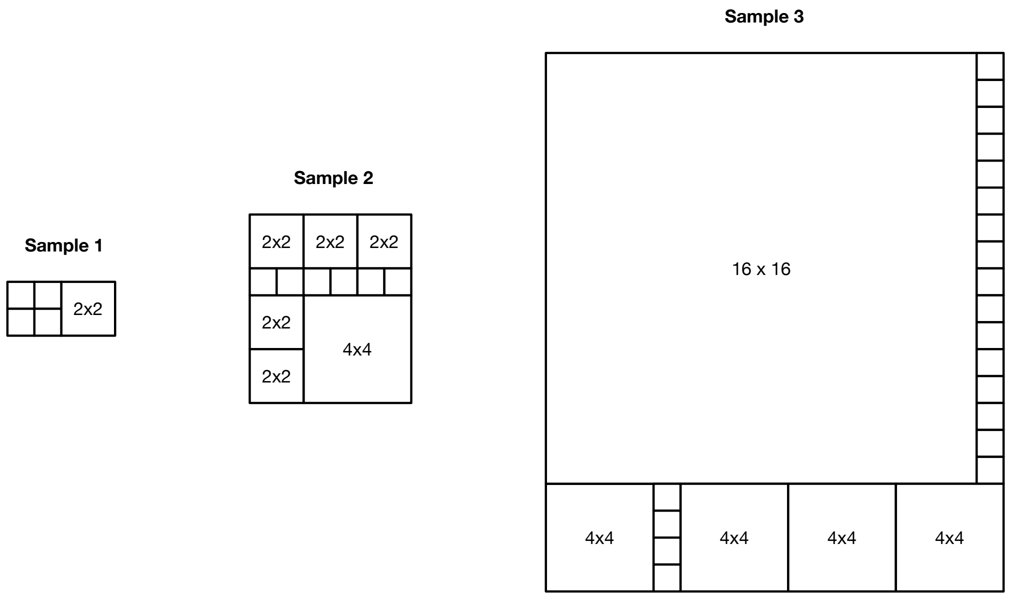

Figure I.1: Optimal square tilings for the first three sample inputs. The small unlabelled tiles are 1×1 tiles.

Input

The first line of input contains three integers W, H and N ($1 \leq W, H \leq 2^{50}$ and $1 \leq N \leq 51$). Here, W and H indicate the dimensions of the rectangle. The next line contains N integers $a_0, a_1, \ldots, a_{N-1}$ where a_i ($0 \leq a_i \leq 2^{51}$) is the number of $2^i \times 2^i$ squares you own.

Output

If there is a square tiling of a $W \times H$ rectangle using the squares you own, output the minimum number of squares needed in such a square tiling. Otherwise, output -1 if there is no square tiling of the rectangle using the squares you own.

Sample Input 1	Sample Output 1
4 2 2 5 1	5

Sample Input 2	Sample Output 2
6 7 3 10 10 10	12

Sample Input 3	Sample Output 3
17 20 5 20 0 4 0 1	25

Sample Input 4	Sample Output 4
4 2 3 3 1 10	-1

Problem J
Rocky Mountain
Time limit: 3 seconds

The Rocky Mountain Cable (RMC) company is planning to run cables from the top peak of the Rocky Mountains to lower points in the mountain range, so that cable cars can be used to transport tourists to the highest peak. A cable must connect from one of the potential sites to the highest peak in a straight line, but the cable cannot cross any part of the mountain range. However, the cable may coincide with a slope.

In order to serve the most number of tourists, it is desirable to connect the cable from the highest peak to the lowest possible site. Help the company determine the best possible site on the left and the right of the highest peak. If there are ties, choose the leftmost site on the left, and the rightmost site on the right.

The mountain range is specified by N sites (x_i, y_i). One of these sites is the unique highest peak (x_p, y_p) such that $1 < p < N$ and $y_p > y_i$ for all $i \neq p$. Note that the highest peak cannot be the first or the last site. The entire mountain range is described by straight line segments connecting (x_i, y_i) to (x_{i+1}, y_{i+1}) for $1 \leq i < N$, such that $x_i < x_{i+1}$.

Input

The first line of input contains the integer N ($3 \leq N \leq 5 \cdot 10^5$) which is the number of sites. The next N lines each contains two integers x_i and y_i, specifying the N sites. The coordinates satisfy $0 \leq x_i, y_i \leq 10^9$. It is guaranteed that there is a unique highest peak, and that $x_i < x_{i+1}$ for all $1 \leq i < N$.

Output

On the first line, output the coordinates of the best site to the left of the highest peak. On the second line, output the coordinates of the best site to the right of the highest peak.

Sample Input 1	Sample Output 1

5
10 10
20 20
30 40
40 30
50 0
	10 10
	40 30

Sample Input 2	Sample Output 2

5
10 10
20 20
30 30
40 20
50 10
	10 10
	50 10

Sample Input 3

```
10
10 10
20 35
30 20
40 40
50 45
60 40
70 30
80 40
90 20
1000 10
```

Sample Output 3

```
20 35
1000 10
```

Problem K
Sandwich Art
Time limit: 3 seconds

Zig, the alien, has received one-too-many odd looks for eating a PBPB (Peanut Butter Pickle on Baguette) sandwich. Zig has decided to consult you, a 100% real human being, to help him make his next sandwich and better pretend to be human. Zig has already started making his next sandwich. He would like to know whether the sandwich has potential to be a masterpiece or if it is already a disaster, based on whether or not it follows the rules of sandwich art so far.

There are many different genres of sandwich. For instance, there are peanut butter, deli, egg salad, hamburger, hot dog, breakfast, and grilled cheese sandwiches, to name a few. Some sandwich ingredients are unique to one genre of sandwich (like tuna), but there are some ingredients that can be used by many, not all, sandwich genres (like a hoagie).

Sandwich art has several rules:

- Certain ingredients may require other ingredients in the sandwich. For example, bagel sandwiches require cream cheese, and jelly requires peanut butter. Note that the reverse is not necessarily true, you can have cream cheese on a sandwich without a bagel.
- The finished sandwich must not include more than a certain number of ingredients, or else they will be too difficult to eat.

A sandwich is considered *finished* if it contains all required ingredients. A finished sandwich belongs to a genre if the sandwich contains only (not necessarily all) the ingredients in the genre's ingredient list. A finished sandwich is a *masterpiece* if it belongs to at least one of the genres.

Input

The first line of input contains 4 integers N G D M: N is the total number of ingredients in Zig's sandwich so far; G is the number of distinct sandwich genres; D is the number of ingredient dependencies; M is the maximum number of total ingredients.

The next row contains N distinct integers, representing the unique IDs of the ingredients already on Zig's sandwich.

The next G lines each begins with an integer K, the number of ingredients in the genre, followed by K distinct integers, representing the unique IDs of the ingredients in the genre.

The next D lines each contains 2 integers I J ($I \neq J$), denoting that if ingredient I is on the sandwich, then ingredient J must also be on the sandwich. All dependencies are distinct. There may be circular dependencies; for example, a sausage requires a hot dog bun, and a hot dog bun requires a sausage.

All integers are greater than 0 and less than 10^6, and there are no more than 10^6 total integers in the input.

Output

If Zig's sandwich is well on its way to becoming a masterpiece, that is, the sandwich is already finished and a masterpiece or has potential to be a masterpiece by adding more ingredients, output `masterpiece`. Otherwise, output `disaster`.

Sample Input 1

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

Sample Output 1

```
masterpiece
```

Sample Input 2

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

Sample Output 2

```
disaster
```

Sample Input 3

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

Sample Output 3

```
disaster
```

Problem L
Sauna
Time limit: 1 second

Kaisa and her friends are going to the sauna!

Now they have to pick the temperature. Each of them has their own temperature preferences as an inclusive range of acceptable values. Can you help them find the temperatures that everyone is happy with?

Since there might be a large number of possible temperatures, they have asked you to just tell them how many options there are and what the lowest temperature that works for everyone is, as then the sauna will need less time to heat up.

Input

The first line of input contains an integer, N, the number of people in the group ($2 \leq N \leq 2 \cdot 10^5$). The next N lines each contains two integers, a_i and b_i, indicating that the ith person's temperature preference is between a_i and b_i, inclusive. Temperatures are given in millidegrees Celsius, with $0 \leq a_i \leq b_i \leq 2 \cdot 10^5$.

Output

Output two integers: first, the number of different temperatures (in millidegrees Celsius) that fit all preferences; and second, the lowest such value.

If there are no values that fit all preferences, instead output “bad news”.

Explanation for sample 1

There are three possible temperatures that all three people will be happy with: 70003, 70004, 70005.

Sample Input 1

```
3
70000 70005
70003 70010
65000 80000
```

Sample Output 1

```
3 70003
```

Sample Input 2

```
2
70000 70500
80000 80100
```

Sample Output 2

```
bad news
```

This page is intentionally left blank.

Problem M
Space Elevator
Time limit: 2 seconds

After years of material science research, scientists have finally devised a material strong enough to support the long-conceived space elevator! This elevator is anchored to the Earth in geostationary orbit to support efficient vertical transportation through Earth's gravity well.

Significant amounts of engineering, time (and funding!) have been poured into its construction, and the first shipments are now ready to launch! However, with all of the work that has been put in, the scientists overlooked one major factor — the number of incoming requests that they would receive!

In the first few minutes, requests came in from the National Space Agency to send water to the International Space Station (ISS), from the Intercollegiate Coalition for Planetary Comprehension (ICPC) to ship low-gravity probes assembled at the ISS, and from the Astronomy Department to launch a new high altitude satellite measuring solar flares, not to mention that the local weather station asked if a lost high-altitude weather balloon could be retrieved on the way back.

To simplify the process, it has been decided that incoming requests will be aggregated and processed on a weekly basis. With the engineering of the elevator, the time and energy loss depend almost entirely on the distances traveled, with factors like potential energy being almost negligible.

The scientists in charge have turned to the Computer Science departments at their local universities to create a program that will read in the requests for the week and find the minimum vertical distance that the elevator must travel to fulfill all of them.

Note that the elevator effectively has no capacity limit for how many packages it can hold at once since requests to move particularly large objects will be handled separately. Also, the elevator is able to stop where it is at once it has fulfilled all of the requests, so input will also include the starting location of the elevator based on where it stopped the previous week.

Input

The first line of input contains 2 integers, N and H_0, the number of requests and the height the elevator is currently at, respectively ($1 \leq N \leq 10^5$).

The next N lines each contains two integers, u_i and v_i, indicating that there is a request to pickup something at height u_i and drop it off at height v_i. Docking procedures require high precision, so all heights are given as millimeters above sea level. All heights satisfy $0 \leq H_0, u_i, v_i \leq 5 \cdot 10^{11}, u_i \neq v_i$.

Output

Output the minimum distance (in millimeters) that the elevator must travel to fulfill all of the requests.

Sample Input 1

```
1 40
30 60
```

Sample Output 1

```
40
```

Sample Input 2

```
4 60
45 95
55 5
50 10
0 50
```

Sample Output 2

```
155
```

Sample Input 3

```
4 80
100 60
10 20
85 95
55 10
```

Sample Output 3

```
120
```