



ICPC International Collegiate Programming Contest // 2025-2026

# The 2025-2026 ICPC Latin America Contests

## ICPC Latin American Regional Contests – 2025

*November 08, 2025*

### Contest Session

*This problem set contains 12 problems; pages are numbered from 1 to 17.*

*This problem set is used in simultaneous contests with the following participating countries:*

Antigua y Barbuda, Argentina, Bolivia, Brasil, Chile, Colombia, Costa Rica, Cuba, Ecuador, El Salvador, Guatemala, México, Perú, Puerto Rico, República Dominicana, Trinidad y Tobago, Uruguay and Venezuela

## General information

Unless otherwise stated, the following conditions hold for all problems.

### Program name

1. Your solution must be called `codename.c`, `codename.cpp`, `codename.java`, `codename.kt`, `codename.py3`, where *codename* is the capital letter which identifies the problem.

### Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

### Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.

# The 2025-2026 ICPC Latin America Contests

## Problem A – Apple Pie

*Author:* Alejandro Strejilevich de Loma, Argentina

Asdrubal likes writing sequences of integers and eating apple pie. Sometimes he does both things simultaneously. These days he is interested in sequences of a particular type. Given an integer  $N \geq 2$ , he wants to find a sequence such that each set of two different integers between 1 and  $N$  appears exactly once in adjacent positions of the sequence. Besides, no other integers can appear in the sequence.

For instance, for  $N = 2$ , there is a single set of two different integers between 1 and  $N$ , which is the set  $\{1, 2\}$ . Thus, Asdrubal would be satisfied with the sequence  $[1, 2]$ . The only other possibility is the sequence  $[2, 1]$ .

For  $N = 3$ , there are three sets of two different integers between 1 and  $N$ :  $\{1, 2\}$ ,  $\{1, 3\}$  and  $\{2, 3\}$ . In this case a valid sequence is  $[2, 1, 3, 2]$ , since the set  $\{1, 2\}$  only appears in the first two positions, the set  $\{1, 3\}$  only appears in the middle, and the set  $\{2, 3\}$  only appears in the last two positions, with no other integers appearing in the sequence. On the other hand, the sequences  $[2, 1, 3]$ ,  $[2, 1, 3, 2, 1]$  and  $[2, 1, 3, 2, 2]$  are invalid.

Asdrubal thinks he has just written one of his sequences for a certain  $N$ . He was about to check whether the sequence was actually valid, when some apple pie fell on the paper. The apple pie covered zero or more elements of the sequence, splitting the rest into two (possibly empty) parts, one on the left of the apple pie, and one on the right. Given the left and right portions of the sequence not covered by apple pie, you must determine whether the original full sequence could have been valid.

### Input

The first line contains an integer  $N$  ( $2 \leq N \leq 1000$ ).

The next line contains an integer  $P$  ( $0 \leq P \leq 5 \cdot 10^5$ ) indicating the number of integers on the left of the apple pie, followed by those  $P$  integers  $L_1, L_2, \dots, L_P$  ( $1 \leq L_i \leq N$  for  $i = 1, 2, \dots, P$ ).

The next line contains an integer  $Q$  ( $0 \leq Q \leq 5 \cdot 10^5$ ) representing the number of integers on the right of the apple pie, followed by those  $Q$  integers  $R_1, R_2, \dots, R_Q$  ( $1 \leq R_i \leq N$  for  $i = 1, 2, \dots, Q$ ).

Both the left and right portions of the sequence are described from left to right.

### Output

Output a single line with the uppercase letter “Y” if the original full sequence could have been valid, and the uppercase letter “N” otherwise.

Sample input 1	Sample output 1
2 0 0	Y

*Explanation of sample 1:*

In this case  $N = 2$  and the apple pie covered the whole sequence ( $P = Q = 0$ ). Since there is at least a valid sequence for  $N = 2$  (for instance,  $[1, 2]$ ), the sequence written by Asdrubal could have been valid.

Sample input 2	Sample output 2
3 1 2 0	Y

# The 2025-2026 ICPC Latin America Contests

*Explanation of sample 2:*

Here  $N = 3$  and the apple pie covered all but the first element of the sequence ( $P = 1$  and  $Q = 0$ ). Since there is at least a valid sequence for  $N = 3$  that starts with  $L_1 = 2$  (such as  $[2, 1, 3, 2]$ ), it follows that the original full sequence could have been valid.

Sample input 3	Sample output 3
3 2 2 1 2 3 2	Y

*Explanation of sample 3:*

Again  $N = 3$ , but now the apple pie covered zero or more elements in the middle of the sequence. Since there is at least a valid sequence for  $N = 3$  that starts with  $[2, 1, \dots]$ , then have zero or more elements in the middle, and finally ends with  $[\dots, 3, 2]$  (again  $[2, 1, 3, 2]$ ), we know that the sequence written by Asdrubal could have been valid.

Sample input 4	Sample output 4
3 2 2 1 1 3	N

Sample input 5	Sample output 5
3 2 2 1 2 2 1	N

Sample input 6	Sample output 6
3 2 2 1 2 2 2	N

# The 2025-2026 ICPC Latin America Contests

## Problem B – Balanced Balloons

*Author:* Marcos Kolodny, Argentina

A group of students is starting a competitive programming club, and they expect that  $N$  people will join, one by one. When each person joins, they must bring between 1 and  $K$  balloons.

The founders want to make sure things stay fair and decided to impose the following fairness rule: after each person joins, it must be possible to evenly redistribute all balloons collected so far among the current members. In other words, after each step, the total number of balloons so far must be divisible by the current number of people.

Being competitive programmers, they realized that this makes for a nice counting problem, and decided to challenge you. How many sequences of  $N$  balloon contributions (each contribution between 1 and  $K$ ) satisfy the fairness rule at every step?

Two sequences are considered different if they differ at any position.

### Input

The input consists of a single line that contains two integers  $N$  ( $1 \leq N \leq 10^9$ ) and  $K$  ( $1 \leq K \leq 2000$ ), indicating respectively the number of members and the maximum number of balloons each member can bring.

### Output

Output a single line with an integer indicating the number of balloon contribution sequences that satisfy the fairness rule. Because this number can be very large, output the remainder of dividing it by 998244353.

Sample input 1	Sample output 1
3 3	5

*Explanation of sample 1:*

The 5 sequences of balloon contributions respecting the fairness rule are:

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

Sample input 2	Sample output 2
4 1	1

*Explanation of sample 2:*

Everyone must contribute with a single balloon so the only possible sequence is  $[1, 1, 1, 1]$ .

# The 2025-2026 ICPC Latin America Contests

## Problem C – Clean Streets

Author: Roberto Sales, Brasil

One more year of Virada Cultural in São Paulo. The city is full of people, and the streets are full of life. People from all over the city go to the streets to enjoy a night of cultural events and a lot of music.

Oswaldo, an employee at the city hall, was assigned the task of hiring cleaners to clean the  $S$  streets of São Paulo, in no more than  $K$  hours after this big event. He was given by his boss a list of  $N$  cleaners he could hire, numbered from 1 to  $N$ , along with information related to how fast they work and the payment they accept. According to the list, cleaner  $i$  can clean any street in  $H_i$  hours, and accepts as payment anything between  $L_i$  and  $U_i$  for cleaning each street.

Oswaldo must hire a subset  $\mathcal{C}$  of the cleaners that are in the list. For each hired cleaner  $i \in \mathcal{C}$  he must assign  $s_i$  streets for them to clean and a payment  $p_i$  per street, taking into account the following guidelines:

- Each number of streets  $s_i$  must be a positive integer and the sum  $\sum_{i \in \mathcal{C}} s_i$  must be exactly  $S$ , because every street must be cleaned and no street can be cleaned by more than one cleaner.
- Note that each hired cleaner  $i$  will take  $s_i \cdot H_i$  hours to clean their assigned streets. Since the cleaners can work in parallel, the cleaning job will take, in total,  $\max_{i \in \mathcal{C}}(s_i \cdot H_i)$  hours to be completed. This total time must be at most  $K$  hours.
- The payment per street  $p_i$  must be a rational number between  $L_i$  and  $U_i$  (that is,  $L_i \leq p_i \leq U_i$ ).
- To ensure a fair hiring process, the payment per hour of work  $\frac{p_i}{H_i}$  must be the same for all hired cleaners.

Notice that the above restrictions do not apply to the cleaners that are not hired.

Hired cleaner  $i$  will receive  $s_i \cdot p_i$  as payment, being the total payment  $\sum_{i \in \mathcal{C}} s_i \cdot p_i$ . Help Oswaldo determine the minimum total payment for cleaning the streets honoring the given guidelines, or tell him that those requirements cannot be satisfied.

### Input

The first line contains three integers  $N$ ,  $S$  and  $K$  ( $1 \leq N, S \leq 10^5$  and  $1 \leq K \leq 10^9$ ), indicating respectively the number of available cleaners, the number of streets to be cleaned, and in how many hours the job must be completed. Each cleaner is identified by a distinct integer from 1 to  $N$ .

The  $i$ -th of the next  $N$  lines describes cleaner  $i$  with three integers  $H_i$  ( $1 \leq H_i \leq 10^5$ ),  $L_i$  and  $U_i$  ( $1 \leq L_i \leq U_i \leq 100$ ), indicating that the cleaner can clean any street in  $H_i$  hours, and accepts as payment any value in the range  $[L_i, U_i]$ .

### Output

If the given guidelines can be satisfied, output a single line with two positive integers  $x$  and  $y$ , such that  $\frac{x}{y}$  is an irreducible fraction indicating the minimum total payment for cleaning the streets according to those guidelines.

If the requirements cannot be fulfilled, output a line with the character “\*” (asterisk) instead.

<b>Sample input 1</b> 2 15 10 1 4 10 2 2 8	<b>Sample output 1</b> 80 1
<b>Sample input 2</b> 2 7 9 3 4 10 2 2 8	<b>Sample output 2</b> 68 3

# The 2025-2026 ICPC Latin America Contests

Sample input 3	Sample output 3
2 15 10 1 4 10 5 2 8	*

# The 2025-2026 ICPC Latin America Contests

## Problem D – Displaying Decimals

Author: Bruno Monteiro, Brasil

After the BRICS group introduced the International Common Payment Currency (ICPC), all currencies (like real, peso, dollar, etc.) are now defined by their values in ICPCs.

To help people navigate this new world order, you decided to create a website to compute the conversion rates between currencies. The website handles  $N$  currencies, numbered from 1 to  $N$ . Each currency  $i$  has an associated integer value  $A_i$ , indicating that one unit of currency  $i$  is worth  $A_i$  ICPCs.

The website works as follows: the user selects a source currency  $i$ , then they select a target currency  $j$  (it might be that  $j = i$ ), and the website displays the conversion rate between them, that is, the exact decimal representation of  $\frac{A_i}{A_j}$ .

To deal with repeating decimals, you decided to display only the first period (represented by a line above it, when it exists). Thus, the quotient  $\frac{4}{4} = 1$  requires 1 digit to be displayed,  $\frac{4}{3} = 1.\bar{3}$  needs 2 digits,  $\frac{41}{4} = 10.25$  uses 4 digits, and  $\frac{3}{14} = 0.214285\bar{7}$  takes 8 digits.

Your website has been running successfully for quite some time. As part of your performance analytics, you want to compute the expected display size of conversion rate pages.

Given the values  $A_1, A_2, \dots, A_N$ , you must compute the expected number of digits needed to display the quotient  $\frac{A_i}{A_j}$ , when  $i$  and  $j$  are chosen uniformly at random between 1 and  $N$ .

### Input

The first line contains an integer  $N$  ( $1 \leq N \leq 10^5$ ) indicating the number of currencies.

The second line contains  $N$  integers  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^5$  for  $i = 1, 2, \dots, N$ ), where  $A_i$  is the value of one unit of currency  $i$  expressed in ICPCs.

### Output

The expected number of digits needed to display the quotient  $\frac{A_i}{A_j}$ , when  $i$  and  $j$  are chosen uniformly at random between 1 and  $N$ , can be expressed as an irreducible fraction  $P/Q$ , with  $Q$  and  $M = 998244353$  coprime. Let  $Q^{-1}$  be the modular multiplicative inverse of  $Q$  modulo  $M$ , that is,  $Q \cdot Q^{-1} \equiv 1 \pmod{M}$ . Output a single line with the integer  $P \cdot Q^{-1} \bmod M$ .

Sample input 1	Sample output 1
3 15 36 14	332748121

Explanation of sample 1:

The table below shows the number of digits needed to display the quotients row/column, each of them with probability  $\frac{1}{9}$ .

	15	36	14
15	1	4	8
36	2	1	7
14	3	3	1

For example,  $\frac{14}{36} = 0.3\bar{8}$  requires 3 digits. The expected number of digits is  $\frac{30}{9} = \frac{10}{3}$ , that is,  $P = 10$  and  $Q = 3$ . Since  $Q^{-1} = 332748118$ , the final answer to be printed is  $P \cdot Q^{-1} \bmod M = 10 \cdot 332748118 \bmod 998244353 = 332748121$ .



# The 2025-2026 ICPC Latin America Contests

## Problem E – Emergency Rations

*Author:* André Amaral de Sousa, Brasil

A village stores emergency ration boxes. Each box contains a certain number of rations, and the village leader keeps track of how many rations are in each box. Initially, there are no boxes in the supply. Sometimes a new box is added to the supply, or an existing box is removed.

In an emergency, the villagers must choose exactly one of the following actions each day that the emergency lasts:

1. pick one non-empty box and use up all rations in that box, or
2. use one ration from every non-empty box.

The leader is worried about the possibility of rapid consumption. Therefore, after each change to the supply, the leader wants to know how many days it would take before all boxes become empty, assuming that an emergency occurs and the villagers choose actions that empty all boxes in the fewest possible days.

Note that the rations are not actually used; the leader only wants to know the minimum number of days needed to empty all boxes. As running these simulations is not something the leader can do alone, you have been asked to help.

### Input

The first line contains an integer  $Q$  ( $1 \leq Q \leq 3 \cdot 10^5$ ) indicating the number of changes to the supply.

The next line contains  $Q$  signed integers  $X_1, X_2, \dots, X_Q$  ( $1 \leq |X_i| \leq 10^9$  for  $i = 1, 2, \dots, Q$ ) indicating, in chronological order, the addition and removal of boxes in the supply. A positive value  $X_i = +c$  means that a box containing  $c$  rations is added, while a negative value  $X_i = -c$  means that a box containing exactly  $c$  rations is removed.

It is guaranteed that every removal is valid, that is, whenever a removal  $X_i = -c$  occurs, there is at least one box in the supply that contains exactly  $c$  rations.

### Output

Output a single line with  $Q$  integers, such that the  $i$ -th integer indicates the minimum number of days needed to use up all rations right after the  $i$ -th change in the supply.

<b>Sample input 1</b> 8 +1 +2 +2 +7 -2 +10 -1 -2	<b>Sample output 1</b> 1 2 2 3 3 4 3 2
<b>Sample input 2</b> 2 +42 -42	<b>Sample output 2</b> 1 0

# The 2025-2026 ICPC Latin America Contests

## Problem F – Fuzzy Factorization

Author: Luis Santiago Re, Argentina

Finding the prime factorization of big numbers is a challenging task. It is so difficult that the security of almost our entire digital world, from online banking to private messages, is built upon how hard it is. In this problem, you are asked to perform such a factorization, but some relative error is allowed in your answer.

More formally, you are given an integer  $X$ , and you have to provide the prime factorization of any number  $Y = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$  such that

- the relative error of the factorization does not exceed  $10^{-9}$  (that is,  $\frac{|X-Y|}{X} \leq 10^{-9}$ ), and
- each prime factor  $p_i$  of  $Y$  does not exceed  $10^{18}$  (that is,  $p_i \leq 10^{18}$  for  $i = 1, 2, \dots, k$ ).

### Input

The input consists of a single line that contains an integer  $X$  ( $2 \leq X \leq 10^{1000}$ ).

### Output

The first line must contain a positive integer  $k$  indicating the number of different prime factors in the prime factorization of  $Y = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$ .

The  $i$ -th of the next  $k$  lines must contain the two positive integers  $p_i$  and  $e_i$ , representing that  $p_i$  is a prime factor of  $Y$  with multiplicity  $e_i$ .

It can be proven that a valid answer exists under the given constraints. If there are multiple solutions, output any of them.

<b>Sample input 1</b>  520	<b>Sample output 1</b>  3 5 1 2 3 13 1
<b>Sample input 2</b>  1073741825	<b>Sample output 2</b>  1 2 30

*Explanation of sample 2:*

$X = 1073741825$ ,  $Y = 2^{30} = 1073741824$ , and the relative error is  $\frac{|X-Y|}{X} = \frac{1}{1073741825} \leq 10^{-9}$ . Note that there are other valid solutions for this test case.

# The 2025-2026 ICPC Latin America Contests

## Problem G – Gridoland Power Gauge

*Author:* Gabriel Poesia, Brasil

Gridoland is a city consisting of a grid of  $N \times M$  blocks, designed for robust energy distribution. The city's power supply is a dam that can effectively generate an arbitrarily large current, and it is connected to the top-left block, at coordinates  $(1, 1)$ . The mayor is planning to host a big, energy-hungry celebration at the city hall, which is located at coordinates  $(N, M)$ , and she has hired you to help the planning committee.

Each block in Gridoland has its own energy transformer. Each transformer has an initial capacity to pass current, but this capacity changes linearly over time at a possibly different rate for each transformer. To help your job, technicians have determined each transformer's initial capacity  $C_{ij}$  and its rate of change  $R_{ij}$  (which can be positive, negative, or zero). Therefore, you know that the capacity of the transformer on block  $(i, j)$  at minute  $t$  will be  $C_{ij} + t \cdot R_{ij}$ .

Current flows within the city thanks to a number of power cables. All power cables in Gridoland are made of a superconductor material that can pass arbitrarily high current in either direction. The mayor herself handed you the city's connectivity map, containing all pairs of (vertically or horizontally) adjacent blocks that are connected by a power cable.

During the celebration there will be no energy consumption in any block except for the city hall, so all the current entering the energy grid will be used for the celebration. The celebration must happen at most  $K$  minutes in the future, because the mayor's term will end at that time. You must answer the following question to the planning committee: what will be the grid's peak capacity to supply the celebration at an integer number of minutes  $t$  in the range  $[0, K]$ ?

At each moment, the grid's capacity from the city hall's point of view is equal to the maximum amount of current that can flow from the power supply to the city hall. The current entering each transformer is limited by that transformer's capacity, and by the constraint that all of the current that enters a block has to leave (except for the city hall, which during the celebration will consume all energy it receives).

### Input

The first line contains four integers  $N$ ,  $M$ ,  $P$  and  $K$ , ( $2 \leq N, M \leq 300$ ,  $P \geq 0$  and  $0 \leq K \leq 10^{12}$ ). The values  $N$  and  $M$  indicate the city dimensions,  $P$  represents the number of power cables in the city's connectivity map, and  $K$  is the number of minutes left in the mayor's term.

Each of the next  $N$  lines contains  $M$  integers. In the  $i$ -th line, the  $j$ -th integer is  $C_{ij}$ , the initial capacity of the transformer on block  $(i, j)$  ( $0 \leq C_{ij} \leq 10^{12}$  for  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, M$ ).

Each of the next  $N$  lines contains  $M$  integers. In the  $i$ -th line, the  $j$ -th integer is  $R_{ij}$ , the rate of change of the transformer on block  $(i, j)$  ( $-10^6 \leq R_{ij} \leq 10^6$  for  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, M$ ).

Each of the next  $P$  lines describes a bidirectional power cable with four integers  $X_1$ ,  $Y_1$ ,  $X_2$  and  $Y_2$  ( $1 \leq X_1, X_2 \leq N$  and  $1 \leq Y_1, Y_2 \leq M$ ), indicating that the cable connects block  $(X_1, Y_1)$  with block  $(X_2, Y_2)$ .

It is guaranteed that all transformers have a non-negative capacity at every moment in the range  $[0, K]$ , that is,  $C_{ij} + t \cdot R_{ij} \geq 0$  for every block  $(i, j)$  and every  $t \in [0, K]$ .

It is guaranteed that each cable connects a different pair of blocks, and that any two blocks connected by a cable share a side.

### Output

Output a single line with an integer indicating the grid's peak capacity to supply the celebration at an integer number of minutes  $t$  in the range  $[0, K]$ .

# The 2025-2026 ICPC Latin America Contests

Sample input 1	Sample output 1
<pre> 2 2 3 10 5 4 5 6 0 0 0 0 2 1 1 1 1 1 1 2 1 2 2 2 </pre>	<pre> 4 </pre>

*Explanation of sample 1:*

In this case the city is a  $2 \times 2$  grid,  $K = 10$ , and all blocks have a transformer with stable capacity because  $R_{ij} = 0$ . The block  $(1, 1)$  connected to the power supply has a transformer with capacity  $C_{1,1} = 5$ . It is connected to the transformer on block  $(1, 2)$  which has capacity  $C_{1,2} = 4$ . This transformer is connected to the city hall's transformer on block  $(2, 2)$ , which has capacity  $C_{2,2} = 6$ . At any time between  $t = 0$  and  $t = K$  the energy grid can supply the city hall with a maximum current of 4.

Sample input 2	Sample output 2
<pre> 2 3 6 12 25 1 18 10 2 50 0 2 -1 1 0 -2 1 1 1 2 1 2 1 3 1 3 2 3 1 1 2 1 2 1 2 2 2 2 2 3 </pre>	<pre> 14 </pre>

*Explanation of sample 2:*

Here the city is a  $2 \times 3$  grid, with  $K = 12$ . There are two paths for energy to flow from block  $(1, 1)$  to the city hall at  $(2, 3)$ . One of them uses the cables in the top row  $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3)$ , and the other one uses the cables in the bottom row  $(1, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (2, 3)$ . The optimal time for the celebration is at  $t = 6$ , when a current of 12 can pass through the upper path, and an additional current of 2 can pass through the lower path.

Sample input 3	Sample output 3
<pre> 2 2 1 10000000000000 5 4 5 6 0 0 0 0 2 1 1 1 </pre>	<pre> 0 </pre>

# The 2025-2026 ICPC Latin America Contests

## Problem H – Harder Horizons

Author: Paulo Cezar Pereira Costa, Brasil

Lola keeps a long to-do list of tasks she needs to finish. Each task has a difficulty number (the bigger the number, the harder the task), and the tasks must be completed in the order they appear.

The list keeps growing because items are rarely removed. Tired of that, Lola decided to take action: she wants to split her list into several consecutive groups of tasks, where each group represents the work she will do in one day.

To stay motivated, Lola wants to feel that each day is a step forward. So, starting from the second day, the hardest task of that day must be strictly harder than the hardest task from the previous day.

Lola also does not want to do too much in one day, so she wants to spread the work over as many days as possible.

What is the maximum number of days she can split the tasks into under these rules?

### Input

The first line contains an integer  $N$  ( $1 \leq N \leq 10^5$ ) indicating the number of tasks.

The second line contains  $N$  integers  $D_1, D_2, \dots, D_N$  ( $1 \leq D_i \leq 10^5$  for  $i = 1, 2, \dots, N$ ), representing the difficulties of the tasks in the order they appear in the list.

### Output

Output a single line with an integer indicating the maximum number of days Lola can split the tasks into, such that the hardest task in each day is strictly harder than the hardest task in the previous day.

Sample input 1	Sample output 1
5 1 2 1 3 4	4

*Explanation of sample 1:*

One optimal way to split the tasks in 4 days is:

1 | 2 1 | 3 | 4

with strictly increasing difficulties of the hardest tasks  $1 < 2 < 3 < 4$ .

Sample input 2	Sample output 2
6 2 1 4 2 3 5	3

*Explanation of sample 2:*

Here an optimal way to split the list in 3 days is:

2 1 | 4 2 | 3 5

with  $2 < 4 < 5$ .

# The 2025-2026 ICPC Latin America Contests

## Problem I – Infiltration Route

Author: Martín Muñoz, Chile

Tomás is a secret agent attempting to infiltrate the super-secure headquarters of the International Competitive Programming Consortium (ICPC). The building has  $2N$  floors, numbered from 1 to  $2N$ . Starting at floor 1, he must reach floor  $2N$  to steal classified algorithms, and then make a dramatic parachute escape from the roof.

He has obtained  $M$  stolen elevator access codes. Each access code connects two specific floors  $S$  and  $T$  ( $S < T$ ) and can be used to force the elevator to move upward from floor  $S$  to floor  $T$ .

The building has an advanced security system composed of  $N$  sensors. Sensor  $i$  monitors floors  $i$  and  $i + N$  (for  $1 \leq i \leq N$ ). When Tomás uses an elevator code to leave a floor, the monitoring sensor detects the intrusion and permanently seals both floors it monitors. Sealed floors cannot be entered, and codes cannot be used from sealed floors. Thus, he can enter at most one floor from each monitored pair.

Tomás needs your help to guarantee the success of the mission. Your task is to determine an *infiltration route*, which is a sequence of floors  $f_1, f_2, \dots, f_k$ , such that

- $f_1 = 1$ ,
- $f_k = 2N$ ,
- there is an access code that allows moving from floor  $f_i$  to floor  $f_{i+1}$  for  $1 \leq i \leq k - 1$ , and
- no two floors  $f_i$  and  $f_j$  in the sequence (with  $i \neq j$ ) are monitored by the same sensor.

If such an infiltration route exists, output it. Otherwise, indicate that the mission is impossible.

### Input

The first line contains two integers  $N$  ( $1 \leq N \leq 500$ ) and  $M$  ( $1 \leq M \leq 1000$ ), indicating respectively that the building has  $2N$  floors and that Tomás has  $M$  access codes. Floors are identified by distinct integers from 1 to  $2N$ .

Each of the next  $M$  lines describes an access code with two integers  $S$  and  $T$  ( $1 \leq S < T \leq 2N$ ), representing that the access code allows movement from floor  $S$  to floor  $T$ .

### Output

If an infiltration route exists, output two lines. The first line must contain an integer  $k$  indicating the number of floors in the route. The second line must contain  $k$  integers  $f_1, f_2, \dots, f_k$  representing the floors included in the route. If there are multiple solutions, output any of them.

If the mission is impossible, output a line with the character “\*” (asterisk) instead.

<b>Sample input 1</b> 1 1 1 2	<b>Sample output 1</b> *
<b>Sample input 2</b> 4 9 1 2 2 3 3 6 6 7 7 8 1 3 3 7 2 6 6 8	<b>Sample output 2</b> 4 1 3 6 8

## The 2025-2026 ICPC Latin America Contests

Sample input 3	Sample output 3
4 8 1 2 1 3 2 3 2 6 3 7 6 7 6 8 7 8	*

# The 2025-2026 ICPC Latin America Contests

## Problem J – Judgmental Crowd

*Author:* Inés Kereki, Uruguay

After a glorious night of stand-up comedy, a comedian wants to measure their success. Not with surveys, not with likes, but with real crowd noises! Luckily, the mic captured all the reactions during the routine. The audio has been converted into a single string  $S$  full of mashed-up sounds, such as applause, laughs, boos, etc.

The comedian uses a powerful tool for evaluating their show: the Instant Comedian Performance Calculator (ICPC). The ICPC is able to analyze  $S$  and calculate the emotional score of the performance, which reflects how well the comedian did on stage.

The score is based on the occurrence of three specific substrings within the string  $S$ :

- each occurrence of “ha” increases the score by 1,
- each occurrence of “boooo” decreases the score by 1, and
- each occurrence of “bravo” increases the score by 3.

Any other substrings in  $S$  (such as “ahhh”, “woohoo” or random gibberish) are ignored and do not affect the score. Your mission is to simulate the ICPC by calculating the emotional score of the performance.

### Input

The input consists of a single line that contains a string  $S$  ( $1 \leq |S| \leq 1000$ ) made of lowercase letters, representing the sequence of crowd sounds picked up by the mic.

### Output

Output a single line with an integer indicating the emotional score of the performance.

<b>Sample input 1</b> boooohaboooo	<b>Sample output 1</b> -1
<b>Sample input 2</b> brhavo	<b>Sample output 2</b> 1
<b>Sample input 3</b> booo	<b>Sample output 3</b> 0
<b>Sample input 4</b> bravooooooooobraboooooooo	<b>Sample output 4</b> 2
<b>Sample input 5</b> buuuuuuthisshowisawfulweshallgonow	<b>Sample output 5</b> 1

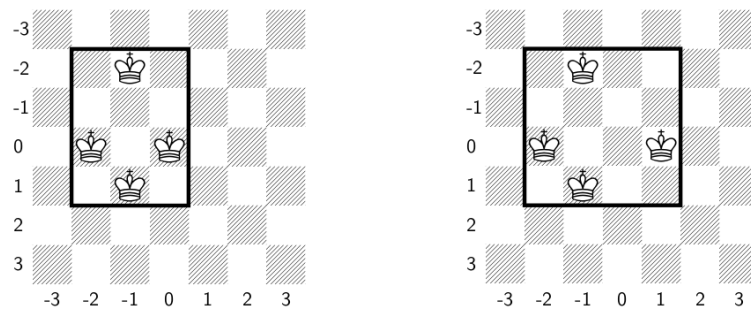


# The 2025-2026 ICPC Latin America Contests

## Problem K – Kings Conquest

Author: Luis Santiago Re, Argentina

There are  $N$  chess kings placed in an infinite grid. Each of them occupies a different cell. The conquered area of the grid is defined as the number of cells in the smallest rectangular subgrid that contains all the cells occupied by the kings. The picture below shows two scenarios; on the left, the conquered area is 12; on the right, it is 16.



You are allowed to perform a sequence of  $K$  moves. In each move, you select any king and move it to any of its eight adjacent cells, as a regular king moves in chess. No two kings may occupy the same cell at the same time.

What is the largest conquered area that can be achieved after performing  $K$  moves?

### Input

The first line contains two integers  $N$  and  $K$  ( $1 \leq N, K \leq 10^5$ ).

Each of the next  $N$  lines contains two integers  $R$  and  $C$  ( $-10^6 \leq R, C \leq 10^6$ ), indicating that a king occupies the cell at row  $R$  and column  $C$ . It is guaranteed that all kings occupy different cells.

### Output

Output a single line with an integer representing the largest conquered area that can be achieved after performing  $K$  moves.

<b>Sample input 1</b> 4 1 1 -1 -2 -1 0 -2 0 0	<b>Sample output 1</b> 16
<b>Sample input 2</b> 2 3 1 1 -1 0	<b>Sample output 2</b> 30
<b>Sample input 3</b> 2 99999 0 0 1 1	<b>Sample output 3</b> 10000200001

# The 2025-2026 ICPC Latin America Contests

## Problem L – Lonely Creatures

Author: Paulo Cezar Pereira Costa, Brasil

There are  $N$  creatures, numbered from 1 to  $N$ , that live in an infinite city represented by the 2D plane. Creature  $i$  lives on the straight line  $y = M_i \cdot x + C_i$ ; it may move anywhere along that line. No two creatures share the same line, but a meeting occurs if their lines intersect. In that case, the intersection point is called a *meeting point* of the two creatures.

The city has just built a new playground. The playground's boundary is the (axis-aligned) parabola  $y = A \cdot x^2 + B$ , where  $A > 0$ . The playground is the open region  $y > A \cdot x^2 + B$ , so points exactly on the boundary are not inside the playground.

Now the city officials want to know whether the new playground is reducing loneliness, and that's where your help is needed. Your task is to count how many distinct pairs of creatures have a meeting point that lies strictly inside the playground. Each pair is counted once; the order of the creatures in a pair does not matter.

### Input

The first line contains three integers  $N$  ( $2 \leq N \leq 10^5$ ),  $A$  ( $1 \leq A \leq 10^4$ ) and  $B$  ( $-10^4 \leq B \leq 10^4$ ), indicating that there are  $N$  creatures and that the playground is the open region of the 2D plane  $y > A \cdot x^2 + B$ . Each creature is identified by a distinct integer from 1 to  $N$ .

The  $i$ -th of the next  $N$  lines contains two integers  $M_i$  and  $C_i$  ( $-10^4 \leq M_i, C_i \leq 10^4$ ), representing that creature  $i$  lives on the straight line  $y = M_i \cdot x + C_i$ . It is guaranteed that no two lines are identical.

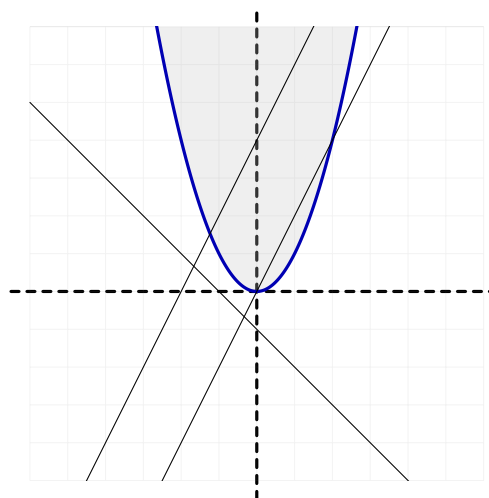
### Output

Output a single line with an integer indicating the number of distinct pairs of creatures whose meeting point lies strictly inside the playground.

Sample input 1	Sample output 1
<pre>3 1 0 -1 -1 2 4 2 0</pre>	<pre>0</pre>

*Explanation of sample 1:*

Creature 1 meets creatures 2 and 3, but in both cases the meeting point lies outside the playground. There are no other meeting points, so the answer is 0.

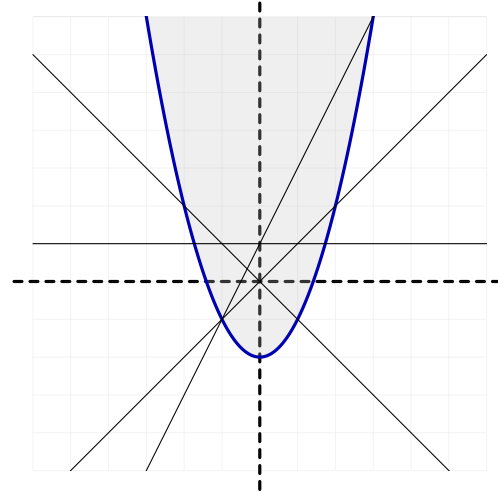


# The 2025-2026 ICPC Latin America Contests

Sample input 2	Sample output 2
<pre>4 1 -2 -1 0 0 1 1 0 2 1</pre>	<pre>5</pre>

*Explanation of sample 2:*

Each pair of creatures have a meeting point. However, the meeting point of creatures 3 and 4 lies on the playground boundary ( $y = A \cdot x^2 + B$ ), so that pair is excluded from the count.



Sample input 3	Sample output 3
<pre>4 1 0 7 -3 5 -2 3 -1 1 0</pre>	<pre>6</pre>