

Problem A. Kolkhozy

Input file: **standard input**
Output file: **standard output**
Time limit: 1.5 seconds
Memory limit: 1024 megabytes

Russia was part of the USSR (Union of Soviet Socialist Republics), which had a socialist regime. Under this regime there were collective farms, such as the kolkhozy (plural of kolkhoz).

Russia used to have n kolkhozy numbered from 1 to n . Each kolkhoz produced k_i bags of grains. This production was split evenly with the m families which were provisioned by that given kolkhoz. According to the strict rules of the Communist Party, when splitting the k_i bags of grains among the m families, each family should get precisely

$$\left\lfloor \frac{k_i}{m} \right\rfloor$$

bags. Note that there may be leftover bags. By law, the leftovers belong to the government. But the strict rules go further. To avoid corruption by the bureaucrats, the government decided that the leftovers from a collective farm go to the x employees in charge of the distribution only if each employee gets exactly one bag.

Unfortunately, when a law is created, hundreds of people try to break it. Thus, the employees decided to split the kolkhozy into regions (arguing that it would simplify distribution) so that they can get the leftover bags.

The bureaucrats considered q possible regions, which are described as intervals from l to r collective farms (endpoints included), and they wish to answer how many kolkhozy in this interval have precisely x leftover bags if this interval were to supply m families.

Input

The first line has two integers n and q - the number of kolkhozy and the number of queries.

The second line has n nonnegative integers, $k_1, \dots, k_n$ — the number of bags of grains produced by each of the n collective farms.

Each one of the next q lines has four integers, l , r , x , and m — the parameters of a query made by the bureaucrats.

Constraints

- $1 \leq n, q \leq 5 \cdot 10^4$
- $0 \leq k_i \leq 5 \cdot 10^4$ for each i
- $1 \leq l \leq r \leq n$
- $1 \leq m \leq 5 \cdot 10^4 + 1$
- $0 \leq x < m$

Output

For each query print a single integer in a single line — how many collective farms in the interval l to r generate exactly x leftover bags of grains when supplying m families.

Example

standard input	standard output
3 4	2
1 2 3	1
1 3 1 2	1
2 3 1 2	3
1 3 0 2	
1 3 0 1	

Problem B. Russo's Russian

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

It is well known that Russian is one of the hardest languages in the world. It is one of the only languages in which declension applies to nouns, adjectives, pronouns and numerals. If you know something about German and have heard of its three complicated declensions, know that there are seven in Russian. Anyway, if you are up to a challenge, try to learn Russian.

What is less known is that there is a Russian variant commonly spoken in Brazil. A Russian family immigrated into Italy in the XVII century. Later, in the wake of the XX century, they came to Brazil, where they live to this day, still preserving their culture and customs. This is the 'Russo' family. Of course, that is not their original surname, but they chose it when mister 'Pshenychnykh', after arriving at Calabria, gave up on trying to teach the Italians how to pronounce 'Pshenychnykh'. Only later he found out that 'Rosso' means red in Italian, but that just made him like the name even more.

Russo's Russian dialect is a mix between Italian, Portuguese, Russian and Calabrian. Its grammar has one of the most complex phrases known to mankind. That became a problem when the descendants of mister Pshenychnykh had to study his finances. Fortunately, the family prospered in Brazil and got really rich. Unfortunately, that created a dispute among the descendants for the wealth and possessions linked to the family's name. Trying to settle things, they went after mister Pshenychnykh's financial notebooks.

Of course, his notes are in the peculiar dialect spoken by the family. Moreover, it is known that mister Pshenychnykh always wrote things respecting the grammar of his dialect — he was really proud of his roots. A note is valid if it a possible expansion of M in the following grammar:

$$\begin{aligned} M &= H \mid P \mid ' \mid M \mid P \\ H &= M \mid \$ \\ P &= P \mid : \mid T \mid T \\ T &= \{ \mid M \mid \} \mid I \end{aligned}$$

As usual, the vertical bar represents possible options for the expansion of a symbol. The characters inside single quotes are the terminal symbols, meaning they are not expanded and appear as they are in the final expression. The symbol I represents any sequence of numerical digits without space. In any other rule, i.e., except in I, it is possible to have any amount of white space separating symbols as described by the rules of the grammar. For example,

{ 12 }

is a valid expression.

Unfortunately, when the years went by, some of the notes got deteriorated, and not all of the expressions remain correct. Your task is to identify which are still correct, i.e., which are strings that can be obtained expanding M in the grammar above.

Input

The input has a single line with a string s with at most 10^5 characters. Every character can be a whitespace, a digit, or one among the following:

$\$: | \{ \}$

Output

Print "YES" if the line supplied is valid, and "NO" otherwise.

Examples

standard input	standard output
1	YES
: 1	NO
\$ 2	YES
5 : 14	YES

Problem C. Crystal Matryoshkas

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

Dollmaking is a well-known Russian tradition. They are often made of wood, and they nest into each other, starting from the largest, which remains in the exterior, to the smallest one, the only one that is not hollow.



The construction of matryoshkas goes back to 1890 by wood carver Vasily Zvyozdochkin, using a design by Sergey Malyutin. In 1900, the dolls were on a world-level exposition in Paris, and they got the bronze medal. After that, they were popularized throughout Russia, and now they can be found in several sizes, themes, and materials. The matryoshkas which are rarest and hardest to build are crystal matryoshkas, conceived by the famous artist Evgeny Kazantsev Swarovsky. Crystal matryoshkas are extremely delicate and built to be put inside the others.

In this matryoshkas, the weights of the dolls play a key role. Swarovsky found that it was only possible to place inside a doll, others dolls whose sum of weights is smaller than or equal to the weight of the most external doll.

The artist needs your help to manage his growing collection. In particular, he must be able to easily record newly created matryoshkas, to record that a matryoshka got broken, or to estimate the size of a matryoshka to be sold — for the purpose of budgeting.

Initially, there are N matryoshka with weights $w_1, \dots, w_N$. You must code a program that, considering the initial collection, responds to Q operations of three possible types:

- $+ W \text{ ID}$ means that Swarovsky build a matryoshka of weight W e identifier ID .
- $- \text{ID}$ means that Swarovsky broke the matryoshka with identifier ID .
- $? \text{ID}$ means that Swarovsky wants to know the size of the largest matryoshka that contains the matryoshka with identifier ID .

All operations will be consistent, that is, there will be no instant when two matryoshka have the same identifier and there will be no operations $-$ or $?$ with invalid matryoshka identifiers.

Input

The first line has two integers, N, Q , which indicate, respectively, the number of matryoshkas already build and the number of operations you must perform.

The next line has N integers, $w_1, \dots, w_N$. The i th number, w_i , indicates the weight of the matryoshka with identifier i , which is in the initial collection.

Each one of the next Q line has an operation of the type $+$, $-$, or $?$, as described above.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq Q \leq 5 \times 10^5$
- $1 \leq W, ID \leq 10^9$

Output

For each query of type $?$ ID print a single line, with the size of the largest matryoshka that contains the matryoshka with identifier ID.

Examples

standard input	standard output
4 4 3 1 2 1 ? 3 + 4 5 ? 3 ? 1	3 4 3
5 6 5 9 4 1 10 ? 2 - 1 - 4 ? 2 + 13 1 ? 2	3 2 3

Note

Explanation for first example

For the first query $? 3$ the answer is 3, since it is possible to form the sequence 1 1 2, which has three matryoshkas and has the matryoshka with identifier 3 (and weight 2).

The second operation is $+ 4 5$, it builds a matryoshka of weight 4 and identifier 5.

After this there is a second query, $? 3$, whose answer is 4, since “1 1 2 4” is a valid matryoshka.

Note that the queried matryoshka is not the most external one in the answer.

The final query is $? 1$, to which the answer is 3, since the largest stacks that include the matryoshka with identifier e (and weight 3) are 1 1 3 or 1 2 3.

Explanation for second example

First there is a query with matryoshka with identifier 2 and weight 9, and answer to which is 3, and there is more than one possible combination: 1 4 9, 1 5 9, and 1 9 10.

After this there is the removal of matryoshkas with identifiers 1 and 4.

Then there is again the query $? 2$, the answer to which is now 2, with possible solutions 4 9 and 9 10.

Then there is an added matryoshka with identifier 1 and weight 13. Note that this operation is consistent since at this point there is no matryoshka with identifier 1.

Finally, the query ? 2 is repeated, and the answer is 3, which is the size of the combination 4 9 13.

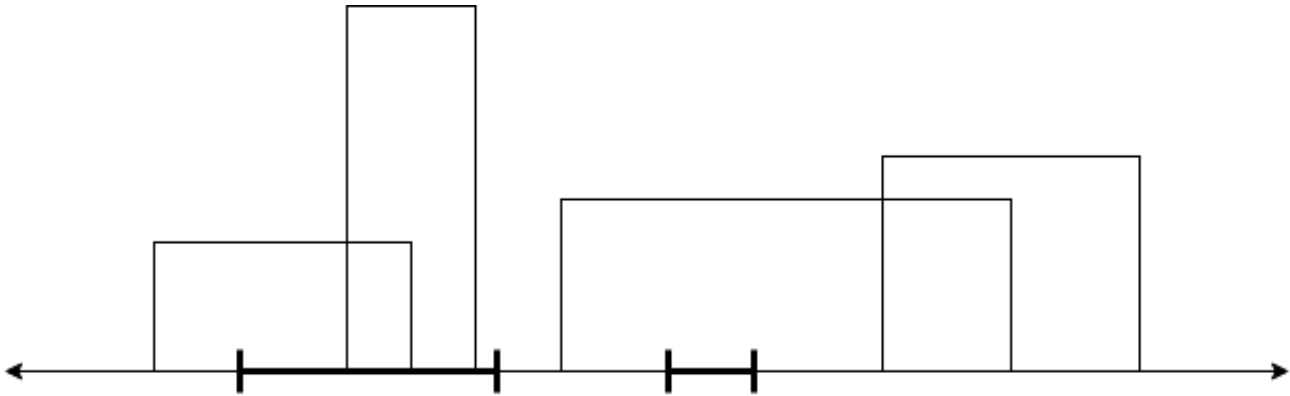
Problem D. Buildings and rockets

Input file: standard input
Output file: standard output
Time limit: 3.5 seconds
Memory limit: 256 megabytes

Throughout history, Russian people have built many buildings in Moscow. Cathedral, theaters, hotels, amazing and expensive buildings which must not be damaged by any means. Their conservation would not be a problem if Russia was not one of the world leaders in rocket technology.

The launch pad is extremely close to the city, and the Russian people are worried about the possibility of collision. To avoid such a tragedy, there is a law which determines the minimum height in which the rockets must be launched. As opposed to most laws, this one is dynamic and changes every time a new building is built.

The city can be seen as a line, with the buildings as rectangles above it. The rocket trajectories can be seen as segments in this line. The safe height for a rocket is defined as the height of the tallest rectangle which intersects its trajectory. Hence, to find the minimum height a rocket must reach, it is enough to find the highest building in the path of the rocket. Note that there may be no buildings in a given path.



As the Russians do not have time for programming, they have trusted you with the task of writing a program that calculates the safe height for every launch, and which adapts itself according to the growth of the city. Good luck!

Input

The first line has an integer n which indicates the amount of events your program must deal with. Each of the following lines describe an event, which can be of two kinds: a new building has been built, or a rocket must be launch. Initially, the city has no buildings at all.

If the event is a new building, the line will be of the form **B** a_s b_s a_t b_t h , indicating that there is a new building which starts at $\frac{a_s \oplus v}{b_s \oplus v}$ (inclusive), ends at $\frac{a_t \oplus v}{b_t \oplus v}$ (inclusive), and has height $h \oplus v$, where v is the last output your program has printed. ($v = 0$ if nothing has been printed).

If the event is a rocket launch, the line will be of the form **S** a_s b_s a_t b_t , indicating there is a rocket whose trajectory starts at $\frac{a_s \oplus v}{b_s \oplus v}$ (inclusive) and ends at $\frac{a_t \oplus v}{b_t \oplus v}$ (inclusive), where v is the last output your program has printed. (Once again, $v = 0$ if nothing has been printed).

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq a_s, b_s, a_t, b_t, h \leq 2^{32}$
- a_s, b_s, a_t, b_t, h are integers
- It is guaranteed that not segment is empty or degenerated

Note: $\oplus$ stands for the bitwise XOR.

Output

For every rocket launch, print in the standard output the safe height for the launch of this rocket, considering the state of the city up until the launch.

Examples

standard input	standard output
6 B 1 2 3 2 4 S 1 2 101 200 S 7 6 297 204 S 5 5 97 96 S 14 5 15 5 S 0 1 1 4	4 4 4 0 0
16 B 17 20 79 23 100 S 4 10 19 21 S 88 119 0 115 B 66 113 75 112 76 S 67 113 73 112 B 66 113 75 112 218 S 67 113 73 112 S 142 170 228 169 S 218 114 130 113 B 70 23 90 22 40 S 80 23 100 1 B 34 60 59 60 164 S 58 60 53 60 S 158 152 164 153 S 173 170 191 191 S 141 141 154 153	100 100 100 190 100 0 40 140 190 140 100

Problem E. Lenin's great dream

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Lenin is one of the most important figures in Russian communism, having his own mausoleum in the Red Square, in Moscow. Following the big events happening recently in Moscow, as the 2018 World Cup and the ACM-ICPC World final in 2020, several organizations are invested in finding the historical secrets of the Russian culture.

The Association of Communists in Moscow (ACM) hired the well known historian Gaia “the curious” Fontenelle to investigate some notebooks which are believed to be written by Lenin. In these notes there are plans and ideas about the future of the Soviet Union. Gaia is interested in the following report.

Lenin wishes to unite all the Soviet Republics into one Republic, which he calls R_1 . Initially, the soviet union is made of $N - 1$ republics: $R_2, R_3, \dots, R_N$. The republic R_i , for $i = 2, 3, \dots, N$, is made of i cities and some (two way) streets among them. The cities inside the republic R_i are identified by the integers $1, 2, \dots, i$. As the Russians are extremely efficient, we can be sure that

- (a) The cities inside a republic are minimally connected. In other words, for every pair of cities there is a unique sequence of streets which connects them.
- (b) Most of the republics were designed in such a way that there is a street between city 1 and every other city.
- (c) The above condition **may not be valid** for only two of the republics, R_a and R_b .

In the notes found there are some remarks about the creation of republic R_1 :

- The republic R_1 will have N cities, identified $1, 2, \dots, N$.
- Between every pair of cities in R_1 there will be a street.
- Each republic R_i , with $i = 2, \dots, N$, will have each of its cities reallocated into a different city in R_1 . Note that, even though two cities of the same republic will not be reallocated into the same city in R_1 , two cities from different republics can be reallocated into the same city in R_1 .
- Every city in R_1 is big enough to receive any number of cities from the former republics.

Despite the fact that Lenin wished to gather everyone in the republic R_1 , he knew that some of the republics had quarrels with each other, and he would like to avoid conflict. Hence, he decided that every street in R_1 would belong to at most one of the old republics, and created the following rule:

- Let x and y be cities that are connected by a street in R_i , for some $i = 2, \dots, N$. If x and y are reallocated into the cities x' and y' (in R_1), then the street that connects x' to y' belongs to the republic R_i .

Gaia found also a map with the description of the republics $R_2, R_3, \dots, R_N$. She wonders whether such grandiose creation as the republic R_1 is possible, and asks for your help.

Input

The first line of the input has 3 integers, N , a and b , separated by spaces, which respectively represent the number of cities in the republic R_1 and the size of the two republics mentioned in the condition (c) of the statement above.

The following $a - 1$ lines has two integers each, u and v , indicating there exists a street that connects the cities u and v in the republic R_a . In these lines, it always holds that $1 \leq u, v \leq a$.

Likewise, the following $b - 1$ lines has two integers each, u and v , indicating there exists a street that connects the cities u and v in the republic R_b . In these lines, it always holds that $1 \leq u, v \leq b$.

Constraints

- $3 \leq N \leq 2500$
- $2 \leq a < b \leq N$
- All the republics involved are minimally connected.

Output

If it is impossible to unite the republics, print **N**. If it is possible, print **Y**, followed by $N - 1$ lines describing a way to reallocate each of the $R_2, R_3, \dots, R_N$ republics. A description of a way to reallocate a republic R_i is a sequence of i integers a_j , with $1 \leq a_j \leq N$, where a_j indicates to which city of R_1 the j th city in R_i was reallocated.

Output

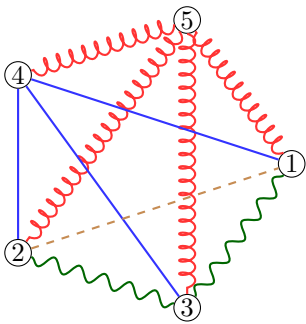
In the first line print ‘Y’ if it is possible to pack $\mathcal{T}$ into K_N , otherwise print ‘N’. In case the packing is possible, the following $N - 1$ describe a desired packing (if multiple packings are possible, print any of them). The i -th of those lines describe the packing of T_i by a sequence of i integers, say $f_1, \dots, f_i$. The integer f_i indicates that in the packing, the vertex j of T_i is mapped to the vertex f_j of K_N .

Examples

standard input	standard output
5 4 5 1 2 1 3 1 4 1 2 1 3 1 4 1 5	Y 2 1 3 1 2 4 1 2 3 5 1 2 3 4
4 3 4 1 2 2 3 1 2 2 3 3 4	Y 2 1 2 4 3 4 1 3 2

Note

We have a picture of the solution described in the first test case. The numbers identify the cities in the republic R_1 , and the streets of the old republics are shown by the different kinds of lines.



Problem F. Forbechenko v Rodvsky

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Two friends from Yuri Gagárin Elementary School, bored with the trivial content of their math lectures, decided to play a game while their third grade class were being taught ordinary differential equations. Vladimir Rodvsky proposed Oleg Forbechenko the following challenge:

Given integers A and B , Forbechenko should write a numerical representation of $\frac{A}{B}$. Forbechenko accepted this promptly, however he later realized Rodvsky's mischief: for some values of A and B it is impossible to represent the fraction in base 10 with a finite number of digits.

For instance, if $A = 1$ and $B = 3$, in base 10 the fraction $\frac{1}{3}$ is represented as

0.333333...

Forbechenko would have to write down infinitely many 3's to complete Rodvsky's challenge!

On second thoughts, Forbechenko remember that the same fraction can be written in base 3 as 0.1. He decided to code a program to find the smallest base $\beta \geq 2$ such that the fraction $\frac{A}{B}$ has a finite representation.

Input

A single input line has two integers A and B , with $1 \leq A, B \leq 10^{18}$ separated by a single space.

Output

Print the single integer β , the smallest base in which it is possible to write down the fraction $\frac{A}{B}$.

Examples

standard input	standard output
1 3	3
3 4	2

Problem G. Hunting leshys

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Leshys are forest spirits that inhabit many regions in Russia, especially in Siberia. Their appearance is almost always masculine and humanoid, but it may appear as an animal or even a mushroom. Sometimes it is depicted with wings and tail, like a demon; other times, as someone familiar to the person with whom he is interacting. His size and social rank can vary independently of his shape and appearance.



Some stories describe the leshy as having wife and kids, like a typical peasant. A single forest may have various leshys, and they form a hierarchical society: each leshy (except for the leshy who is the leader in that forest) has a single leshy to whom he reports.

Leshys were created at the beginning of time, but they do not know each other. At that time, each one believed to be the leader of the forest. It is only when a leshy meets another one that they compare their hierarchies and decide who is inferior to whom.

Leshys are also very lazy. When a leshy sees a threat in the forest, he must report the threat to his superior, which in turns reports to its superior, and so on, until the news reach some leshy who believes to be the leader of the forest. When all the leshy are aware of the threat, the one in charge tasks the least powerful one to deal with the problem. Note that hierarchy and power are distinct attributes for leshys.

In this exercise, you will code a program that simulates a society of leshys. There are n leshys, numbered from 1 to n , and an integer that indicates their power. Initially, each one believes himself to be the leader of the forest.

After that there are m operations, which may be one of two types:

$+ i j$ This indicates that leshys i and j have met, and now leshy j is below leshy i in the hierarchy.

$? i$ This indicates that leshy i identified a threat in the forest. In this case, your program must answer with the power of the leshy that must deal with this threat.

Input

The first line has two integers n and m .

After this there are n integers in the second line, $p_1, p_2, \dots, p_n$. The i th such value is p_i , which indicates the power of leshy i .

After this there are m integers indicating an operation of two leshys meeting or the identification of a threat.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 5 \times 10^5$
- $0 \leq p_i \leq 10^9$
- $1 \leq i, j \leq n$
- You can assume that leshy hierarchies never forms a cycle.

Output

Your output must have an integer for each operation of type ? i , indicating the power of the leshy that should respond to the threat to the forest.

Examples

standard input	standard output
4 6 3 12 5 6 ? 2 + 3 2 ? 2 ? 3 + 4 1 ? 1	12 5 5 3
5 5 5 2 2 1 9 + 4 2 + 3 1 + 5 3 + 1 4 ? 1	2

Problem H. Course recommendation

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

One of the biggest challenges in an online course platform is to recommend to a student one of its over 800 courses.

For instance, João took courses on Java and Java for the Web, with grades 10 and 9, respectively. What should he be recommended next? If we recommend him something much easier, much harder, or simply something that does not go with his profile, we will waste his time. Just as an undergraduate student is very careful for choosing the appropriate courses and not waste a semester due to poor choices, we wish to help students in this platform to make the best choices.

In this problem, we have N students and M courses. Students are numbered from 1 to N , the courses from 1 to M . For each student, we know which courses he or she already took and the respective grades. To recommend a course to a student, we first find another student who is closest to him or her. The distance between two students is defined as the euclidean distance between the grades of the courses both students took. If there are no shared courses, the distance is defined as infinity. Among the courses already taken by the closest student, we recommend the course with the highest grade which was not already taken by the original student. In case of a tie, recommend the course with the smallest index. If the original student already took all the courses that were taken by his closest student, no recommendation is made.

Your task is to code a program that recommend a course to all the students in the platform.

Input

The first line has two integers N and M , the number of students and courses, respectively. After this there are N blocks of lines, each one contains a line with an integer Q_i , the amount of courses already taken by the i th student, and then Q_i lines, each one containing two integers c and g , which indicate that the i th student got grade g for course c .

Constraints

- $2 \leq N \leq 100$
- $1 \leq M \leq 100$
- $1 \leq Q_i, c \leq M$
- $0 \leq g \leq 10$
- You may assume that each student has already taken at least one course that another student took.

Output

Print N lines, where the i th line has the course recommended to the i th student. If student i has already taken all courses that his closest student already took, print -1 .

Examples

standard input	standard output
2 3 2 1 10 2 10 2 2 9 3 5	3 1
2 4 3 1 7 2 8 3 10 3 4 10 2 9 1 5	4 3

Note

In the first example, the distance between students is given by $\sqrt{(10 - 9)^2}$

In the second, it is given by $\sqrt{(7 - 5)^2 + (8 - 9)^2}$

Problem I. Sobytiynny Proyekt Casino

Input file: `standard input`
Output file: `standard output`
Time limit: 0.5 seconds
Memory limit: 256 megabytes

Sobytiynny Proyekt Casino is located in San Petersburg, one of the most beautiful cities in Russia, known (besides being where ITMO is located) by its delightful Hermitage Museum. The casino's owner, a former ICPC contestant who never made it to the world finals, is creating a new game (as the traditional games are not appreciated there...) and he wants revenge on his fellow competitors, by punishing players who are trying to play his game optimally.

In each round, the players get a pair of integers (p, f) , where $p > 0$. The player gets p rubles and he or she may decide to leave the game by paying f rubles (or receiving $-f$ rubles, if f is negative). Depending on how they play, players may end the game while owing money...

Note also that if the player reaches the last pair, he is forced to pay f_N .

Help the casino's owner by finding an ordering of the N pairs which minimize the gains from a player who plays optimally. Your code must print how much the player would earn if she played in the ordering of pairs that minimizes her gain.

Input

You should read the number N of pairs in the first line of input.

After this N lines will follow, each with two numbers separated by a space, p and f .

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq p_i \leq 10^9$ for all i
- $-10^9 \leq f_i \leq 10^9$ for all i

Output

Print a single integer: how much an optimal player would gain in an ordering of pairs that minimizes her gain.

Examples

standard input	standard output
3 1 2 3 3 2 1	3
1 1 -2	3
2 0 1 3 4	-1

Note

Let us analyze the possible permutations of the sequence of pairs for the first case:

Both permutations $(1, 2), (3, 3), (2, 1)$ and $(3, 3), (1, 2), (2, 1)$ award 5 to an optimal player. The permutations $(2, 1), (3, 3), (1, 2)$ and $(3, 3), (2, 1), (1, 2)$ award at most 4.

Finally, the permutations $(1, 2), (2, 1), (3, 3)$ and $(2, 1), (1, 2), (3, 3)$ award at most 3 to a player. Thus, 3 is the smallest value that an optimal player can get in this game.

Problem J. MasterCodeChef Russia

Input file: `standard input`
Output file: `standard output`
Time limit: 3 seconds
Memory limit: 256 megabytes

The success of Russian competitors in programming contests caught the attention of Russian TV channels. Channel Russia 1, in trying to capitalize on the fact that the ACM-ICPC 2020 will be held in Moscow, decided to create a *reality show* on competitive programming, the MasterCodeChef Russia.

However, as soon as they realized the format of the competition, the producers were concerned. How to keep the TV show captivating for 5 hours, in this world of 280 characters and 10-minute videos?

So they made two decisions. The first was to adapt the test, and the second was to hire the legendary *talk show host* Carlos Eduardo Ferreira, from Brazil.

To plan the show, the producers also needed to understand how competitive programmers work. After many field studies, they concluded that a problem p may be described by its theoretical difficulty p_t and by its coding (or implementation) difficulty p_i . A good way to summarize a problem is by its total difficulty $d(p)$, defined by

$$d(p) = p_t^2 + p_i^2.$$

Similarly, a contestant c may be described by the difficulties he imposes into himself when trying to solve a problem: the theoretical difficulty c_t and its coding difficulty c_i . We may summarize its total difficulty $s(c)$ by

$$s(c) = c_t^2 + c_i^2.$$

The producers concluded that a contestant c can solve a problem p within

$$T(c, p) = \frac{s(c) + d(p)}{2}$$

seconds.

In view of this, producers decided that each program will have just one question for the contestants. All contestants start to solve the problem at the same time, and the contest ends when the last contestant finishes his problem.

The beginning of a competition is the most boring moment for spectators. All the competitors are with their hands off the keyboards, gazing at the screen and figuring out what to do. A contestant c solving a problem p only gets interesting in the last $I(c, p) = c_t p_t + c_i p_i$ seconds of the time he takes to solve the problem, and, of course, after solving the problem, since he starts giving interviews to the reporters and joins the rest of the show.

Since every spectator has a favorite contestant and the producers want everyone to be happy, they will use their secret weapon: The amazing entertainment skills of their show host Carlinhos. Carlinhos has to entertain the public until the last contestant gets interesting, so that no spectator gets bored and changes channel.

Since Carlinhos is such a star, it is extremely expensive to pay for his air time. Hence, the producers want to minimize the time t in which he has to entertain the spectators. To do so, they want to choose a problem p such that the inequality

$$T(c, p) - I(c, p) \leq t$$

holds for every contestant c , with minimal t .

Input

The input starts with a positive integer $0 < n \leq 10^5$. After this there are n lines, each with two positive reals c_t and c_i , which describe the theoretical and coding skills of the i th contestant.

For each contestant, we have $0 \leq c_t \leq 10^4$ and $0 \leq c_i \leq 10^4$.

Output

Print 3 real numbers p_t , p_i , and t , which are the theoretical difficulty of the chosen problem, the coding difficulty, and the amount of time during which Carlinhos needs to speak.

We will accept as correct answers with absolute or relative error below 10^{-4} .

Examples

standard input	standard output
2 0 0 4 0	2 0 2
5 0 0 0 1 1 0 1 1 2 2	1 1 1

Problem K. Poor Folk

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Russian writers are known worldwide. One of the main Russian authors is Fyódor Dostoyévskiy. The influence of his works outgrew literature itself and reached other areas such as Philosophy, Psychology, and Theology. His first successful book was “Poor Folk”, published in 1846.



This broad and impacting influence is due in part to the deep psychological description of his characters. However, some have criticized him for writing stories which are too long and even weary. Even though he is considered a great writer, Dostoyévskiy’s life was not easy. In many occasions he had to write under pressure to make a living and perhaps his prolificacy came from such moments of needs.

Dostoyévskiy’s only way to obtain money is by selling chapters of his books. Moreover, he knows the editor would buy each and every chapter he wrote — after all, he is a great writer. However, he wants more than just selling all the chapters he is writing. He noted that his financial woes are caused by his lack of financial discipline. That is why Dostoyévskiy decided that whenever he needs to pay his debts, he will choose some chapters to sell so that the money earned is exactly the amount owed. In this way, he will never have any savings and he will have no need to be disciplined about spending.

Of course he understands there could be some debts he will not be able to pay if he insists on this rule. Thus, knowing the value in rubles for each chapter he already wrote, one of the questions he raised was: what is the smallest debt that he would not be able to pay with previously written chapters? Your task is to answer this question.

Input

You should read an integer n in the first line, the number of chapters already written by Dostoyévskiy. The second line has n integers, $p_1, p_2, \dots, p_n$, the price in rubles for each of the n chapters.

Constraints

- $1 \leq n \leq 5 \cdot 10^5$.
- $1 \leq p_i \leq 10^{12}$, for all i .

Output

A single positive integer, the value of the smallest amount of debt as described above.

Examples

standard input	standard output
3 2 1 100000000000	4
1 2	1