The 45th ACM International Collegiate Programming Contest

Asia Yinchuan Regional Contest

Asia Regional 2020

icpc International Collegiate Programming Contest

宁夏理工学院

Ningxia Institute of Science and Technology

2021 年 5 月 16 日

Problem A. Best Player

Input file:	`standard input`
Output file:	`standard output`

Though Bob has finally returned to the college, he has to stay in his dorm with the other three roommates. Don't know who proposes.

"Let's play a game!"

It was meant to be light-hearted, and the rules are as follows:

- A roommate is designated in turn, and he writes down on paper the coordinates of n points in three-dimensional Euclidean space.
- Each of the other roommates selects one of the X-axis, the Y-axis, and the Z-axis, but no two of them make the same choice. After that, each of them counts the number of points after projecting all written points along the direction of the chosen axis. Under the projection, some points may not be distinguishable. Points $(1, 2, 1)$ and $(1, 2, 5)$ projected along the direction of the Z-axis, for instance, are not distinguishable.
- The roommate with the most points counted will win the game. In case of a tie, the one with the most points counted whose selected axis has the smallest character in the alphabetical order will be the winner. Note that in the alphabetical order X is less than Y, Y is less than Z, and by transitivity, X is less than Z.

Bob does want to win.

"Please help me; I entreat you."

Add flowers to brocade people, all over finish is; Give timely assistance and have several people?

Input

The first line contains an integer n $(1 \le n \le 100)$.

In the next n lines, each line contains three integers x, y and z $(-100 \le x, y, z \le 100)$, representing a three-dimensional point at (x, y, z).

Output

Output a character t $(t \in \{X, Y, Z\})$ in a line indicating that the winner's choice is the t-axis. Note that the output characters are case-sensitive.

Examples

standard input	standard output
2 1 1 1 1 2 1	X
3 1 2 9 1 2 2 2 2 9	Y
4 -100 -100 100 -100 100 100 100 -100 100 100 100 100	Z

Problem B. The Great Wall

Input file:	`standard input`
Output file:	`standard output`

Beacon towers are built throughout and alongside the Great Wall. There was once a time when there were n beacon towers built from west to east for defending against the invaders. The altitude of the i-th beacon tower, based on historical records, is a_i.

The defenders divide strategically all beacon towers into k parts where each part contains several, but at least one, consecutive beacon towers. The scale of an individual part is given by the difference between the highest and the lowest altitudes of beacon towers, and the most sensible partition maximizes the sum of scales of all parts.

As a historian, you are dying to know the maximum sums of scales for every $k = 1, 2, \ldots, n$.

Input

The first line contains an integer n ($1 \le n \le 10^4$), denoting the number of beacon towers alongside the Great Wall.

The second line contains n integers $a_1, a_2, \ldots, a_n$, where the i-th integer a_i ($1 \le a_i \le 10^5$) is the altitude of the i-th beacon tower.

Output

Output n lines, the i-th of which contains an integer indicating the maximum sum for $k = i$.

Examples

standard input	standard output
5 1 2 3 4 5	4 3 2 1 0
5 1 2 1 2 1	1 2 2 1 0

Problem C. Lucky Sequence

Input file: `standard input`
Output file: `standard output`

A number sequence $[a_1, a_2, \dots, a_n]$ is lucky if and only if the following requirements are fulfilled.

- Each element a_i in the sequence is a non-negative integer such that $0 \le \frac{a_i}{i} < \frac{\sqrt{5}+1}{2}$;
- For any two elements a_i and a_j $(i \ne j)$ in the sequence, $a_i \ne 0$ and $a_j \ne 0$ imply that $a_i \ne a_j$.

Your task is to figure out how many number sequences of length n are lucky and report the number modulo 998 244 353.

Input

The first line contains an integer T $(1 \le T \le 10^5)$, indicating the number of test cases.

Then follow T test cases. For each test case:

The only line contains an integer n $(1 \le n \le 10^5)$, indicating the length of the sequence.

Output

For each test case, output an integer in one line, representing the number of lucky sequences of length n modulo 998 244 353.

Example

standard input	standard output
5 1 2 3 4 5	2 7 27 141 919

Note

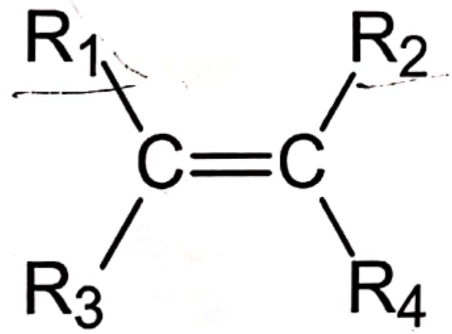
For $n = 2$, there are 7 lucky sequences: $[0,0]$, $[0,1]$, $[0,2]$, $[0,3]$, $[1,0]$, $[1,2]$ and $[1,3]$.

Problem E. Isomerism

Input file: `standard input`
Output file: `standard output`

In chemistry, isomerism is the phenomenon in which more than one compounds have the same chemical formula but different chemical structures. Chemical compounds that have identical chemical formulae but differ in properties and the arrangement of atoms in the molecule are called isomers.

Ethylene, which has a carbon-carbon double bond, is one of the most important fundamental chemicals in the petrochemical industry as it is the source material for a variety of products such as polyethene resin, ethylene glycol, vinyl chloride resin, acetic acid, styrene, and alpha-olefin which are produced by polymerization, oxidation, alkylation, hydration, or the addition of halogen. The precise format of an ethylene derivative is given in the figure below, where R_1, R_2, R_3, R_4 are atoms or atomic groups. We always suppose that R_1 and R_2 are on the same side of the carbon-carbon double bond, while R_3 and R_4 are on the other side. The carbon-carbon double bond in an ethylene derivative cannot rotate around the bond axis.



To distinguish isomers of the ethylene derivatives, two different naming methods, say Cis-Trans isomerism and Zusammen-Entgegen isomerism, are invented in the academic circle. The different scopes of application between these two methods are listed as follows:

- If a carbon atom connects with two identical atoms or atomic groups, isomerism of the given ethylene derivative does not exist; otherwise
- if some atoms or atomic groups connecting with carbon atoms are the same, the ethylene derivative is called Cis-Trans isomerism. If the two identical atoms or atomic groups lie on the same side (i.e. upside or downside in the figure above) of the carbon-carbon double bond, it is called Cis-isomerism, or else it is called Trans-isomerism;
- if the four atoms or atomic groups connecting with carbon atoms are pairwise distinct, the ethylene derivative is called Zusammen-Entgegen isomerism. If the atom or the atomic group of R_1 and R_3 with a higher priority and the atom or the atomic group of R_2 and R_4 with a higher priority lie on the same side (i.e. upside or downside in the figure above) of the carbon-carbon double bond, it is called Zasamman-isomerism, or else it is called Entgegen-isomerism.

All the atoms or atomic groups which may appear in R_1, R_2, R_3 and R_4 are listed as follows in descending order of the priority, the first of which is the one with the highest priority.

- `-F, -Cl, -Br, -I, -CH3, -CH2CH3, -CH2CHCH3, -H`

Now, you are asked to determine if there is any isomerism for a given ethylene derivative and find out the naming method it fits for when possible.

Input

The first line contains an integer T ($1 \le T \le 10^5$), indicating the number of test cases.

Then follow T test cases. For each test case:

The only line contains four strings R_1, R_2, R_3 and R_4 ($R_1, R_2, R_3, R_4 \in$ {`-F`, `-Cl`, `-Br`, `-I`, `-CH3`, `-CH2CH3`, `-CH2CHCH3`, `-H`}), which are the atoms or atomic groups connecting to carbon atoms of the ethylene derivative.

Output

For each test case, output a string in one line, describing the type of isomerism the ethylene derivative fits for as follows:

- If there is no isomerism of this ethylene derivative, output "`None`";
- If it is Cis-isomerism, output "`Cis`";
- If it is Trans-isomerism, output "`Trans`";
- If it is Zasamman-isomerism, output "`Zasamman`";
- Otherwise, it should be Entgegen-isomerism, so output "`Entgegen`".

Example

standard input	standard output
2 -H -H -H -Cl -F -F -Br -Cl	None Cis

Problem F. Maximize the Ratio

Input file: `standard input`
Output file: `standard output`

In geometry, a convex polygon is a simple polygon (which is not self-intersecting) in which no segment joining a point to another one on the boundary goes outside the polygon. Equivalently, it is a simple polygon whose interior is a convex set.

Given n distinct points on the plane, you may draw some segments such that:

- At least one segment is drawn.
- The length of each segment is positive and finite.
- Each endpoint of any segment belongs to the given set of points.
- For any two segments, they either share an endpoint or have no intersection.
- All segments form a convex polygon.

Let the area of the polygon be A, and let the sum of the squares of the lengths of segments be B. Your task is to maximize the ratio of A to B.

Input

The first line contains an integer T ($1 \le T \le 10$), indicating the number of test cases. Then T test cases are following.

For each test case: the first line contains an integer n ($3 \le n \le 500$) indicating the number of given points. In the next n lines, each line contains two integers x and y ($-10^4 \le x, y \le 10^4$) representing a point at (x, y).

For all points in an individual test case, it is guaranteed that no two points are identical and no three points are collinear. It is also guaranteed that the sum of n in all test cases is up to 500.

Output

For each test case, output the maximal ratio of A to B in a single line. Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}.

Formally speaking, suppose that your output is a and the jury's answer is b. Your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
2 4 0 0 0 5 5 5 5 0 4 0 0 0 5 5 0 2 2	0.250000000000000 0.125000000000000

Problem G. Photograph

Input file: `standard input`
Output file: `standard output`

Graduation season is approaching and it's time to take commemorative photos.

There are n students in Bob's class, each assigned with a unique student number from 1 to n, where the height of the student numbered i is h_i. They want to take photos at the school gate and q more memorable places. For each place, students will arrive in a known order, and, as they hope for more personal engagement, once a student arrived they take a new photo so that n photos will be taken. After finishing all the n photos at the place, students will go back to their dorms and set up a new time for the next place.

To make the queue orderly, every time students have to stand in a row in ascending order of their student numbers when taking a new photo, and the orderliness of the photo is defined as the sum of the squares of the height differences between every two neighbouring students in the row. Your task is to, for each place, calculate the sum of the orderlinesses of all the n photos that are taken there.

Input

The first line contains two integers n and q ($1 \le n \le 10^5, 0 \le q \le 100$), indicating the number of students and the number of memorable places.

The second line contains n integers indicating the heights of students, the i-th of which is h_i ($1 \le h_i \le 10^4$).

The third line contains n pairwise distinct integers $p_1, p_2, \ldots, p_n$, ranged from 1 to n, representing the student numbers in the order of their attendance at the school gate.

In the next q lines, the i-th line contains an integer k ($0 \le k \le 10^9$), which says that the order of attendance for students at the $(i+1)$-th place is rescheduled from the previous order by shifting left $(k + lastans)$ times, where the $lastans$ is the sum of the orderlinesses of the n photos taken at the i-th place.

Note that the order $p_1, p_2, \ldots, p_{n-1}, p_n$ after shifting left once will turn to the new order $p_2, p_3, \ldots, p_n, p_1$.

Output

Output $(q+1)$ lines, where the i-th line contains an integer representing the sum of the orderlinesses of the n photos taken at the i-th place.

Example

standard input	standard output
5 4 1 2 3 4 5 1 2 3 4 5 6 6 8 10	10 10 13 21 36

Note

For the sample case, the order of students' attendance at each place are listed as follows:

$[1, 2, 3, 4, 5] \rightarrow [2, 3, 4, 5, 1] \rightarrow [3, 4, 5, 1, 2] \rightarrow [4, 5, 1, 2, 3] \rightarrow [5, 1, 2, 3, 4]$.

Problem H. Absolute Space

Input file: standard input
Output file: standard output

Xiao Ming is an elementary school student. One day his math teacher instructed the students to create a series of special models, which are called "absolute space". Each model consists of many stars in the three-dimensional space and is associated with a special parameter n claiming that for any star in the model, there exist exactly n stars whose Euclidean distance to it is 1.

For example, in the case of $n = 1$, you can create this model by drawing a line of length 1, and then drawing the stars at the line's endpoints. Similarly, for $n = 2$, you can create this model by drawing a regular polygon (i.e. equilateral triangle, square and regular pentagon) with side length 1, and then drawing the stars at the polygon's vertices. As you can see, there are infinite kinds of "absolute space".

In this class, the teacher taught them what kind of possibilities look like in the case of $n = 1, 2$, and their homework was to create a model for $n = 3$. Since the homework is too easy, Xiao Ming doesn't want to stop at these small cases. He spent a whole night reflecting on that and finally managed to find solutions applicable for any parameter n ($= 1, 2, 3, 4, 5, \ldots$). Furthermore, he found ways to shrink the number of stars when creating these models.

Such great progress brought him ecstasy, as well as insomnia. When he got up the next morning, he completely forgot how he got the idea, and his draft model had also collapsed due to the out-of-date glue he mistakenly used, so he has no proof to convince his teacher about what he has done. Could you please create a model for this poor little boy?

Given the parameter n, your task is to create any possible model such that the number of stars is not greater than a certain limit.

Input

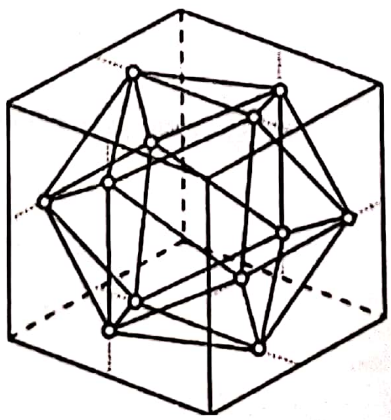
The only line contains an integer n ($1 \le n \le 10$), indicating the parameter of the model.

Output

In the first line, output an integer m ($1 \le m \le 100$), indicating the number of stars in your model.

In the next m lines, output the coordinates of each star in one line, that is, output three real numbers x, y and z ($-100 \le x, y, z \le 100$) in one line, representing a star at point (x, y, z). Besides, it is not allowed for any two points in your output to have a Euclidean distance less than 10^{-6}.

To tolerate the precision error, your model will be regarded as valid if, for any star in your model, there exist exactly n stars whose Euclidean distance to it is in the range $(1 - 10^{-6}, 1 + 10^{-6})$.



Examples

standard input	standard output
1	2 0.000000000 0.000000000 0.000000000 1.000000000 0.000000000 0.000000000
2	3 0.000000000 0.000000000 0.000000000 1.000000000 0.000000000 0.000000000 0.500000000 0.866025404 0.000000000
3	4 0.500000000 0.000000000 -0.353553391 -0.500000000 0.000000000 -0.353553391 0.000000000 0.500000000 0.353553391 0.000000000 -0.500000000 0.353553391
4	6 0.707106781 0.000000000 0.000000000 -0.707106781 0.000000000 0.000000000 0.000000000 0.707106781 0.000000000 0.000000000 -0.707106781 0.000000000 0.000000000 0.000000000 0.707106781 0.000000000 0.000000000 -0.707106781
5	12 0.000000000 0.500000000 0.809016994 0.000000000 -0.500000000 0.809016994 0.000000000 0.500000000 -0.809016994 0.000000000 -0.500000000 -0.809016994 0.500000000 0.809016994 0.000000000 0.500000000 -0.809016994 0.000000000 -0.500000000 0.809016994 0.000000000 -0.500000000 -0.809016994 0.000000000 0.809016994 0.000000000 0.500000000 0.809016994 0.000000000 -0.500000000 -0.809016994 0.000000000 0.500000000 -0.809016994 0.000000000 -0.500000000

Note

For the sample cases:

- A segment of length 1 is a feasible model for $n = 1$;
- an equilateral triangle is a feasible model for $n = 2$;
- a regular tetrahedron is a feasible model for $n = 3$;
- a regular octahedron is a feasible model for $n = 4$;
- and a regular icosahedron is a feasible model for $n = 5$ shown in the figure above.

Problem I. The Answer!

Input file: `standard input`
Output file: `standard output`

Millions of years ago, a very smart hyperspace race built a supercomputer, DeepThought. They gave DeepThought two positive integers x and y, and let her calculate **The Answer** to Life, the Universe and Everything.

DeepThought didn't know how to calculate **The Answer** and she was in a hurry to watch TV, so she made a big integer using very complicated steps and no one would know how she got the result:

- Firstly, DeepThought chose an integer a greater than 1.
- Secondly, she calculated $(a^{F_x} - 1)$ and $(a^{F_y} - 1)$, denoted by u and v respectively, where F_n is the n-th term of the Fibonacci sequence, given by the recursion: $F_1 = 1$, $F_2 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for integer $n \geq 3$.
- Lastly, she computed the ratio of these two numbers' least common multiple $\text{lcm}(u, v)$ to their greatest common divisor $\gcd(u, v)$ as **The Answer**, which is obviously an integer.

Since she has gone for leisure activities, the task to calculate **The Answer** is left to you. For the secrecy of **The Answer**, you are only asked to report **The Answer** modulo m.

Input

The first line contains an integer T $(1 \leq T \leq 10^4)$, indicating the number of test cases.

Then follow T test cases. For each test case:

The only line contains four integers x, y, a and m $(1 \leq x, y \leq 10^9, 2 \leq a, m \leq 10^9)$.

Output

For each test case, output an integer in one line, representing **The Answer** modulo m.

Example

standard input	standard output
3 3 3 3 97 7 3 2 1901 6 12 3 100	1 1761 98

Note

For the first example case, $F_x = F_y = 2$, $u = v = a^2 - 1 = 8$, $\text{lcm}(u, v) = \gcd(u, v) = 8$, so **The Answer** is $8/8 = 1$, and you need to report $1 \bmod 97 = 1$.

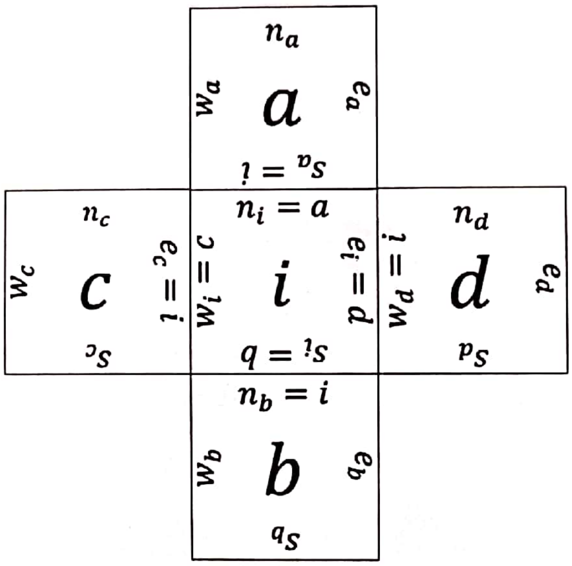
For the second example case, $F_x = 13$, $F_y = 2$, $u = a^{13} - 1 = 8191$, $v = a^2 - 1 = 3$, $\text{lcm}(u, v) = 24573$, $\gcd(u, v) = 1$, so **The Answer** is $24573/1 = 24573$, and you need to report $24573 \bmod 1901 = 1761$.

Problem J. Let's Play Jigsaw Puzzles!

Input file: `standard input`
Output file: `standard output`

Bob bought a new jigsaw puzzle yesterday. The puzzle consists of m^2 rectangular pieces, where each piece contains a unique number ranged from 1 to m^2 used for identification and four associated numbers indicating the adjacent pieces.

Specifically, the piece numbered i is described with four numbers n_i, s_i, w_i and e_i, corresponding to four adjacent pieces: the piece numbered n_i (resp., s_i, w_i and e_i) lie on the north side (resp., south side, east side and west side). If an associated number is -1, no more piece would lie on the corresponding side.



Undoubtedly, the goal of this game is to arrange all pieces in the correct locations. The instruction says that the complete image is square, with m rows and m columns.

At this juncture, Bob is at the International Collegiate Programming Contest in Yinchuan. He has no time to solve the jigsaw puzzle but has to solve Problem K instead. Can you help him?

Input

The first line contains an integer m ($1 \le m \le 10^3$).

In the next m^2 lines, the i-th line contains four integers n_i, s_i, w_i and e_i ($n_i, s_i, w_i, e_i \in \{-1\} \cup [1, m^2]$). It is guaranteed that a corresponding solution does exist.

Output

Output a sorted jigsaw puzzle in m lines from north to south, where each line contains m integers representing the pieces from west to east in the corresponding row of the jigsaw puzzle.

Example

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

Problem K. Browser Games

Input file: `standard input`
Output file: `standard output`

In the upcoming n days, n browser games will be released on a new website. According to the plan, the administrator will release a new game per day. Users have to open the corresponding URL (Uniform Resource Locator) and get feedback from the server to download a game.

However, the setup of the server uses unreadable legacy codes. Once a user somehow finds the URL of an unreleased game, the data of the game would leak out. To temporarily fix the problem, the administrator decided to add a series of confirmation prefixes, which are non-empty strings, at the server-side. The server will respond with the correct game data when the requested URL does correspond to a game (no matter released or unreleased) and at least one confirmation prefix is a prefix of the URL; otherwise, the server will declare that the game is not found.

To make the work easier, the administrator asks you to find the minimum number of confirmation prefixes the server required to avoid data leaks every time after a new game release.

Input

The first line contains an integer n ($1 \le n \le 5 \times 10^4$), indicating the number of browser games to be released.

In the next n lines, the i-th line contains a non-empty string, consisting of only lowercase letters ('a' to 'z'), dots ('.') and forward slashes ('/'), indicating the URL of the browser game released on the i-th day.

It is guaranteed that the length of each given URL is at most 50, and no given URL is the prefix of any other given URL.

Output

Output n lines, the i-th of which contains an integer indicating the minimum number of required confirmation prefixes after the i-th new game released.

Example

standard input	standard output
3 ufoipv.ofu hsbocmvfgboubtz.kq hfotijo.njipzp.dpn/kb	1 2 2

Problem L. Sheep Village

Input file: `standard input`
Output file: `standard output`

There is an old country but called Sheep Village which contains n cities numbered from 1 to n and m bidirectional roads, each of which connects two different cities.

In Sheep Village, cities are connected through roads. That is, you can always find a path from a city to any other city through some roads. Besides, each road here belongs to at most one simple circuit, where a simple circuit is a set of roads that forms a cyclic path $u_1 \to u_2 \to \ldots \to u_m \to u_1$ $(m \geq 1)$ without passing a city more than once. Note that the cyclic paths $a \to b \to c \to a$, $b \to c \to a \to b$ and $a \to c \to b \to a$ correspond to the same circuit.

There are k sheep living in Sheep Village and also k lurking wolves. Once all sheep fall asleep, the lurking wolves, led by the wolf king, will launch a blitzkrieg for their static prey. Quietly running through a road does cost energy. For the sake of energy-saving, the wolf king hopes for the best assignments for each wolf to catch a distinct sheep such that the total energy consumed in catching sheep is as small as possible.

As a brilliant strategist as well as a wolf, it's time for you to make the decision to meet the king's requirement.

Input

The first line contains three integers n, m and k $(2 \leq n \leq 10^5, n-1 \leq m \leq 2n-2, 1 \leq k \leq 10^5)$, indicating the number of cities in Sheep Village, the number of roads between cities, and the total number of sheep (or wolves) respectively.

The second line contains k integers, of which the i-th number a_i $(1 \leq a_i \leq n)$ indicates the i-th wolf is lurking in the city numbered a_i.

The third line contains k integers, of which the i-th number b_i $(1 \leq b_i \leq n)$ indicates the i-th sheep is sleeping in the city numbered b_i. Some sheep and wolves may live in a city together.

In the next m lines, each line contains three integers u, v and w $(1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 10^5)$ representing a bidirectional road connecting the cities numbered u and v that costs w energy for an individual wolf running through it quietly. There may exist more than one road between any two cities.

Output

Output an integer in a line representing the minimum total energy consumed.

Example

standard input	standard output
5 8 4 2 2 3 3 4 4 5 5 1 2 1 2 1 1 1 3 1 3 1 1 1 4 1 4 1 1 1 5 1 5 1 1	8

Problem M. Tower of the Sorcerer

Input file:	`standard input`
Output file:	`standard output`

It may seem that the story is usual. A warrior is trying to save the princess caught in the tower. But this is not a usual ordinary story. Well, this story is even not from this game.

Tower of the Sorcerer appears to be RPG-style; you have a strength (STR) and a health bar showing the health points (HP), and so do all the monsters. When you engage a monster, you and the monster take turns hitting each other until one falls, and you go first. Each attack will cause damages that make the health point of the one being attacked decreased by the strength of the attacker. A character falls if its health point drops to zero or lower after an attack.

As an excellent hacker and a player manipulating the warrior, you set your initial health point as infinity. There are n monsters inside the tower. You can fight against the monsters in any order. After defeating a monster, you can set your strength to the strength of the defeated monster.

What is the minimum total amount of damages you will receive?

Input

The first line contains two integers n ($1 \le n \le 10^5$) and STR_w ($1 \le STR_w \le 10^5$), indicating the number of monsters and the initial strength of the warrior.

In the next n lines, the i-th line contains two integers STR_i and HP_i ($1 \le STR_i, HP_i \le 10^5$), representing the strength and the health point of the i-th monster.

Output

Output an integer in a line, representing the minimum total amount of damages the warrior will receive.

Example

standard input	standard output
4 1 3 2 4 4 5 6 1 6	9

Note

For the sample case:

1. You, the warrior, attack the first monster; the monster receives 1 damage. The first monster starts to attack you, causing 3 damages. You attack the first monster once again; it receives 1 more damage and falls. Then you set your strength to 3.
2. You attack the fourth monster; the monster receives 3 damages. The fourth monster starts to attack you, causing 1 damage. You attack the fourth monster once again; it receives 3 more damages and falls.
3. You attack the third monster; it receives 3 damages. The third monster starts to attack you, causing 5 damages. You attack the third monster once again; it receives 3 more damages and falls. Then you set your strength to 5.
4. You attack the second monster; it receives 5 damages and falls. Then the game ends, and you receives 9 damages in total.