Problem A
Clock Breaking

After numerous unfortunate freak fatalities and the lawsuits, settlements, protests, and boycotts that naturally followed, the beleaguered executives at ACME Clock Manufacturers have decided they need to finally fix their disastrous quality control issues. It has been known for years that the digital clocks they manufacture have an unacceptably high ratio of faulty liquid-crystal display (LCD) screens, and yet these heartless souls have repeatedly failed to address the issue, or even warn their hapless consumers!

You have been called in as a quality consultant to finally put a stop to the madness. Your job is to write an automated program that can test a clock and find faults in its display.

These clocks use a standard 7-segment LCD display for all digits (shown on the left in Figure A.1), plus two small segments for the ':', and show all times in a 24-hour format. The minute before midnight is 23:59, and midnight is 0:00. The ':' segments of a working clock are on at all times. The representation of each digit using the seven segments is shown on the right in Figure A.1.

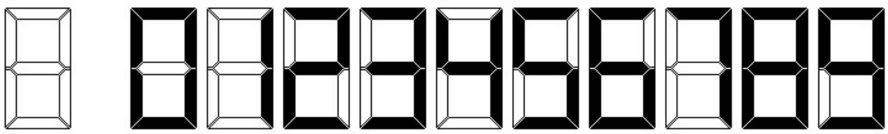


Figure A.1: LCD display of each digit.

Your program will be given the display of a clock at several consecutive minutes, although you do not know exactly what time these displays start. Some of the LCD segments are burnt out (permanently off) and some are burnt in (permanently on). Your program must determine, where possible, which segments are definitely malfunctioning and which are definitely in working order.

Input

The first input line contains a single integer n ($1 \leq n \leq 100$), which is the number of consecutive minutes of a clock's display. The next $8n - 1$ lines contain n ASCII images of these clock displays of size 7×21, with a single blank line separating the representations.

All digit segments are represented by two characters, and each colon segment is represented by one character. The character '`X`' indicates a segment that is on. The character '`.`' indicates anything else (segments that are off or non-segment portions of the display). See the sample input/output for details; the first output shows every possible LCD segment along with the smaller segments used to represent the ':'. No clock representation has an '`X`' in a non-segment position or only half of a segment showing.

Output

Display a 7×21 ASCII image with a '`0`' for every segment that is burnt out, a '`1`' for every segment that is burnt in, a '`W`' for every segment that is definitely working, and a '`?`' for every segment for which the status cannot be determined. Use '`.`' for non-segments. If the given displays cannot come from consecutive minutes, display `impossible`.

Sample Input 1

```
3
......XX.....XX...XX.
.....X..X...X..X....X
.....X..X.X.X..X....X
.............XX...XX.
.....X..X......X.X..X
.....X..X......X.X..X
......XX.....XX...XX.

......XX.....XX...XX.
.....X..X...X..X....X
.....X..X.X.X..X....X
.............XX...XX.
.....X..X......X.X..X
.....X..X......X.X..X
......XX.....XX...XX.

.............XX...XX.
........X...X..X....X
........X.X.X..X....X
.............XX......
........X...X..X.X..X
........X...X..X.X..X
......XX.....XX...XX.
```

Sample Output 1

```
.??...WW.....??...??.
?..?.W..?...?..1.0..?
?..?.W..?.?.?..1.0..?
.??...??.....11...WW.
?..?.W..?.0.W..?.1..?
?..?.W..?...W..?.1..?
.??...11.....??...??.
```

Sample Input 2

```
2
......XX.....XX...XX.
...X....X...X..X.X..X
...X....X.X.X..X.X..X
......XX..........XX.
...X.X....X.X..X.X..X
...X.X......X..X.X..X
......XX.....XX...XX.

......XX.....XX......
...X....X...X..X.....
...X....X.X.X..X.....
......XX.............
...X.X....X.X..X.....
...X.X......X..X.....
......XX.....XX......
```

Sample Output 2

```
impossible
```

Problem B
Forever Young

My birthday is coming up. Alas, I am getting old and would like to feel young again. Fortunately, I have come up with an excellent way of feeling younger: if I write my age as a number in an appropriately chosen base b, then it appears to be smaller. For instance, suppose my age in base 10 is 32. Written in base 16 it is only 20!

However, I cannot choose an arbitrary base when doing this. If my age written in base b contains digits other than 0 to 9, then it will be obvious that I am cheating, which defeats the purpose. In addition, if my age written in base b is too small then it would again be obvious that I am cheating.

Given my age y and a lower bound ℓ on how small I want my age to appear, find the largest base b such that y written in base b contains only decimal digits, and is at least ℓ when interpreted as a number in base 10.

Input

The input consists of a single line containing two base 10 integers y ($10 \le y \le 10^{18}$ – yes, I am *very* old) and ℓ ($10 \le \ell \le y$).

Output

Display the largest base b as described above.

Sample Input 1	Sample Output 1
`32 20`	`16`

Sample Input 2	Sample Output 2
`2016 100`	`42`

This page is intentionally left blank.

Problem C
Metal Processing Plant

Picture from Wikimedia Commons

Yulia works for a metal processing plant in Ekaterinburg. This plant processes ores mined in the Ural mountains, extracting precious metals such as chalcopyrite, platinum and gold from the ores. Every month the plant receives n shipments of unprocessed ore. Yulia needs to partition these shipments into two groups based on their similarity. Then, each group is sent to one of two ore processing buildings of the plant.

To perform this partitioning, Yulia first calculates a numeric distance $d(i, j)$ for each pair of shipments $1 \leq i \leq n$ and $1 \leq j \leq n$, where the smaller the distance, the more similar the shipments i and j are. For a subset $S \subseteq \{1, \ldots, n\}$ of shipments, she then defines the *disparity* D of S as the maximum distance between a pair of shipments in the subset, that is,

$$D(S) = \max_{i,j \in S} d(i, j).$$

Yulia then partitions the shipments into two subsets A and B in such a way that the sum of their disparities $D(A) + D(B)$ is minimized. Your task is to help her find this partitioning.

Input

The input consists of a single test case. The first line contains an integer n ($1 \leq n \leq 200$) indicating the number of shipments. The following $n - 1$ lines contain the distances $d(i, j)$. The i^{th} of these lines contains $n - i$ integers and the j^{th} integer of that line gives the value of $d(i, i + j)$. The distances are symmetric, so $d(j, i) = d(i, j)$, and the distance of a shipment to itself is 0. All distances are integers between 0 and 10^9 (inclusive).

Output

Display the minimum possible sum of disparities for partitioning the shipments into two groups.

Sample Input 1	Sample Output 1
5 4 5 0 2 1 3 7 2 0 4	4

Sample Input 2

```
7
1 10 5 5 5 5
5 10 5 5 5
100 100 5 5
10 5 5
98 99
3
```

Sample Output 2

```
15
```

Problem D
Oil

A large part of the world economy depends on oil, which is why research into new methods for finding and extracting oil is still active. Profits of oil companies depend in part on how efficiently they can drill for oil. The International Crude Petroleum Consortium (ICPC) hopes that extensive computer simulations will make it easier to determine how to drill oil wells in the best possible way.

Drilling oil wells optimally is getting harder each day – the newly discovered oil deposits often do not form a single body, but are split into many parts. The ICPC is currently concerned with stratified deposits, as illustrated in Figure D.1.

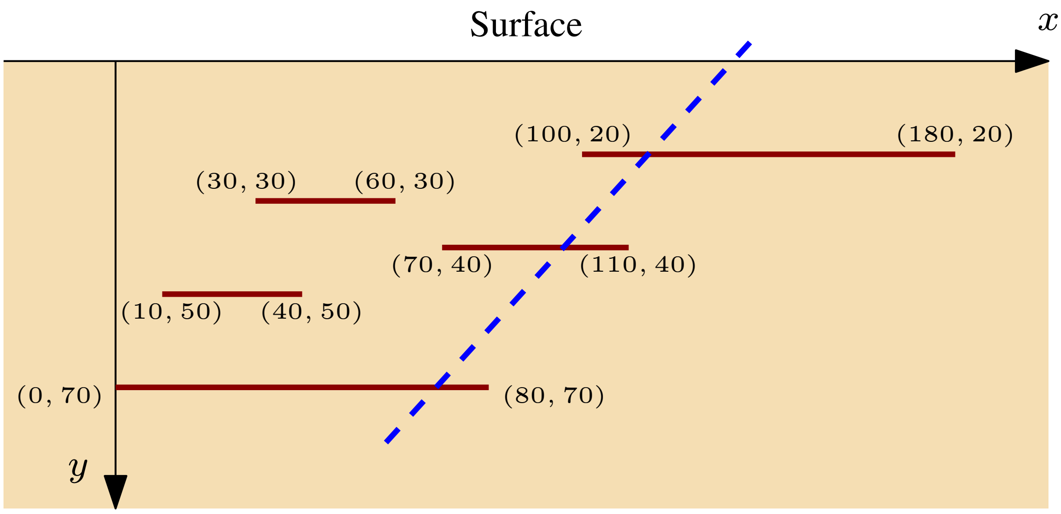


Figure D.1: Oil layers buried in the earth. This figure corresponds to Sample Input 1.

To simplify its analysis, the ICPC considers only the 2-dimensional case, where oil deposits are modeled as horizontal line segments parallel to the earth's surface. The ICPC wants to know how to place a single oil well to extract the maximum amount of oil. The oil well is drilled from the surface along a straight line and can extract oil from all deposits that it intersects on its way down, even if the intersection is at an endpoint of a deposit. One such well is shown as a dashed line in Figure D.1, hitting three deposits. In this simple model the amount of oil contained in a deposit is equal to the width of the deposit. Can you help the ICPC determine the maximum amount of oil that can be extracted by a single well?

Input

The first line of input contains a single integer n ($1 \leq n \leq 2\,000$), which is the number of oil deposits. This is followed by n lines, each describing a single deposit. These lines contain three integers x_0, x_1, and y giving the deposit's position as the line segment with endpoints (x_0, y) and (x_1, y). These numbers satisfy $|x_0|, |x_1| \leq 10^6$ and $1 \leq y \leq 10^6$. No two deposits will intersect, not even at a point.

Output

Display the maximum amount of oil that can be extracted by a single oil well.

Sample Input 1

```
5
100 180 20
30 60 30
70 110 40
10 40 50
0 80 70
```

Sample Output 1

```
200
```

Sample Input 2

```
3
50 60 10
-42 -42 20
25 0 10
```

Sample Output 2

```
25
```

Problem E
Need for Speed

Sheila is a student and she drives a typical student car: it is old, slow, rusty, and falling apart. Recently, the needle on the speedometer fell off. She glued it back on, but she might have placed it at the wrong angle. Thus, when the speedometer reads s, her true speed is $s + c$, where c is an unknown constant (possibly negative).

Sheila made a careful record of a recent journey and wants to use this to compute c. The journey consisted of n segments. In the i^{th} segment she traveled a distance of d_i and the speedometer read s_i for the entire segment. This whole journey took time t. Help Sheila by computing c.

Note that while Sheila's speedometer might have negative readings, her true speed was greater than zero for each segment of the journey.

Input

The first line of input contains two integers n ($1 \leq n \leq 1\,000$), the number of sections in Sheila's journey, and t ($1 \leq t \leq 10^6$), the total time. This is followed by n lines, each describing one segment of Sheila's journey. The i^{th} of these lines contains two integers d_i ($1 \leq d_i \leq 1\,000$) and s_i ($|s_i| \leq 1\,000$), the distance and speedometer reading for the i^{th} segment of the journey. Time is specified in hours, distance in miles, and speed in miles per hour.

Output

Display the constant c in miles per hour. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1	Sample Output 1
3 5 4 -1 4 0 10 3	3.000000000

Sample Input 2	Sample Output 2
4 10 5 3 2 2 3 6 3 1	-0.508653377

This page is intentionally left blank.

Problem F
Swap Space

You administer a large cluster of computers with hard drives that use various file system types to store data. You recently decided to unify the file systems to the same type. That is quite a challenge since all the drives are currently in use, all of them are filled with important data to the limits of their capacities, and you cannot afford to lose any of the data. Moreover, reformatting a drive to use a new file system may significantly change the drive's capacity. To make the reformat possible, you will have to buy an extra hard drive. Obviously, you want to save money by minimizing the size of such extra storage.

You can reformat the drives in any order. Prior to reformatting a drive, you must move all data from that drive to one or more other drives, splitting the data if necessary. After a drive is reformatted, you can immediately start using it to store data from other drives. It is not necessary to put all the data on the same drives they originally started on – in fact, this might even be impossible if some of the drives have smaller capacity with the new file system. It is also allowed for some data to end up on the extra drive.

As an example, suppose you have four drives A, B, C, and D with drive capacities 6, 1, 3, and 3 GB. Under the new file system, the capacities become 6, 7, 5, and 5 GB, respectively. If you buy only 1 GB of extra space, you can move the data from drive B there and then reformat drive B. Now you have 7 GB free on drive B, so you can move the 6 GB from drive A there and reformat drive A. Finally, you move the six total gigabytes from drives C and D to drive A, and reformat C and D.

Input

The input begins with a line containing one integer n ($1 \leq n \leq 10^6$), which is the number of drives in your cluster. Following this are n lines, each describing a drive as two integers a and b, where a is the capacity with the old file system and b is the capacity with the new file system.

All capacities are given in gigabytes and satisfy $1 \leq a, b \leq 10^9$. (One thousand petabytes should be enough for everyone, right?)

Output

Display the total extra capacity in gigabytes you must buy to reformat the drives.

Sample Input 1	Sample Output 1
4 6 6 1 7 3 5 3 5	1

Sample Input 2	Sample Output 2
4 2 2 3 3 5 1 5 10	5

This page is intentionally left blank.

Problem G
Clock Breaking 2

After numerous unfortunate freak fatalities and the lawsuits, settlements, protests, and boycotts that naturally followed, the beleaguered executives at ACME Clock Manufacturers have decided they need to finally fix their disastrous quality control issues. It has been known for years that the digital clocks they manufacture have an unacceptably high ratio of faulty liquid-crystal display (LCD) screens, and yet these heartless souls have repeatedly failed to address the issue, or even warn their hapless consumers!

You have been called in as a quality consultant to finally put a stop to the madness. Your job is to write an automated program that can test a clock and find faults in its display.

These clocks use a standard 7-segment LCD display for all digits (shown on the left in Figure G.1), plus two small segments for the ':', and show all times in a 24-hour format. The minute before midnight is 23:59, and midnight is 0:00. The ':' segments of a working clock are on at all times. The representation of each digit using the seven segments is shown on the right in Figure G.1.

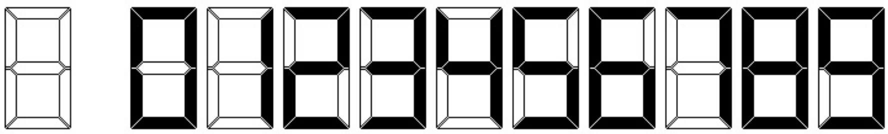


Figure G.1: LCD display of each digit.

Your program will be given the display of a clock at several consecutive minutes, although you do not know exactly what time these displays start. Some of the LCD segments are burnt out (permanently off) and some are burnt in (permanently on). Your program must determine, where possible, which segments are definitely malfunctioning and which are definitely in working order.

Input

The first input line contains a single integer n ($1 \leq n \leq 100$), which is the number of consecutive minutes of a clock's display. The next $8n - 1$ lines contain n ASCII images of these clock displays of size 7×21, with a single blank line separating the representations.

All digit segments are represented by two characters, and each colon segment is represented by one character. The character '`X`' indicates a segment that is on. The character '`.`' indicates anything else (segments that are off or non-segment portions of the display). See the sample input/output for details; the first output shows every possible LCD segment along with the smaller segments used to represent the ':'. No clock representation has an '`X`' in a non-segment position or only half of a segment showing.

Output

Display a 7×21 ASCII image with a '`0`' for every segment that is burnt out, a '`1`' for every segment that is burnt in, a '`W`' for every segment that is definitely working, and a '`?`' for every segment for which the status cannot be determined. Use '`.`' for non-segments. If the given displays cannot come from consecutive minutes, display `impossible`.

Sample Input 1

```
3
......XX.....XX...XX.
.....X..X...X..X....X
.....X..X.X.X..X....X
.............XX...XX.
.....X..X......X.X..X
.....X..X......X.X..X
......XX.....XX...XX.

......XX.....XX...XX.
.....X..X...X..X....X
.....X..X.X.X..X....X
.............XX...XX.
.....X..X......X.X..X
.....X..X......X.X..X
......XX.....XX...XX.

.............XX...XX.
........X...X..X....X
........X.X.X..X....X
.............XX......
........X...X..X.X..X
........X...X..X.X..X
......XX.....XX...XX.
```

Sample Output 1

```
.??...WW.....??...??.
?..?.W..?...?..1.0..?
?..?.W..?.?.?..1.0..?
.??...??.....11...WW.
?..?.W..?.0.W..?.1..?
?..?.W..?...W..?.1..?
.??...11.....??...??.
```

Sample Input 2

```
2
......XX.....XX...XX.
...X....X...X..X.X..X
...X....X.X.X..X.X..X
......XX..........XX.
...X.X....X.X..X.X..X
...X.X......X..X.X..X
......XX.....XX...XX.

......XX.....XX......
...X....X...X..X.....
...X....X.X.X..X.....
......XX.............
...X.X....X.X..X.....
...X.X......X..X.....
......XX.....XX......
```

Sample Output 2

```
impossible
```

Problem H
Forever Young 2

My birthday is coming up. Alas, I am getting old and would like to feel young again. Fortunately, I have come up with an excellent way of feeling younger: if I write my age as a number in an appropriately chosen base b, then it appears to be smaller. For instance, suppose my age in base 10 is 32. Written in base 16 it is only 20!

However, I cannot choose an arbitrary base when doing this. If my age written in base b contains digits other than 0 to 9, then it will be obvious that I am cheating, which defeats the purpose. In addition, if my age written in base b is too small then it would again be obvious that I am cheating.

Given my age y and a lower bound ℓ on how small I want my age to appear, find the largest base b such that y written in base b contains only decimal digits, and is at least ℓ when interpreted as a number in base 10.

Input

The input consists of a single line containing two base 10 integers y ($10 \leq y \leq 10^{18}$ – yes, I am *very* old) and ℓ ($10 \leq \ell \leq y$).

Output

Display the largest base b as described above.

Sample Input 1	Sample Output 1
`32 20`	`16`

Sample Input 2	Sample Output 2
`2016 100`	`42`

This page is intentionally left blank.

Problem I
Metal Processing Plant 2


Picture from Wikimedia Commons

Yulia works for a metal processing plant in Ekaterinburg. This plant processes ores mined in the Ural mountains, extracting precious metals such as chalcopyrite, platinum and gold from the ores. Every month the plant receives n shipments of unprocessed ore. Yulia needs to partition these shipments into two groups based on their similarity. Then, each group is sent to one of two ore processing buildings of the plant.

To perform this partitioning, Yulia first calculates a numeric distance $d(i, j)$ for each pair of shipments $1 \leq i \leq n$ and $1 \leq j \leq n$, where the smaller the distance, the more similar the shipments i and j are. For a subset $S \subseteq \{1, \ldots, n\}$ of shipments, she then defines the *disparity* D of S as the maximum distance between a pair of shipments in the subset, that is,

$$D(S) = \max_{i,j \in S} d(i, j).$$

Yulia then partitions the shipments into two subsets A and B in such a way that the sum of their disparities $D(A) + D(B)$ is minimized. Your task is to help her find this partitioning.

Input

The input consists of a single test case. The first line contains an integer n ($1 \leq n \leq 200$) indicating the number of shipments. The following $n - 1$ lines contain the distances $d(i, j)$. The i^{th} of these lines contains $n - i$ integers and the j^{th} integer of that line gives the value of $d(i, i + j)$. The distances are symmetric, so $d(j, i) = d(i, j)$, and the distance of a shipment to itself is 0. All distances are integers between 0 and 10^9 (inclusive).

Output

Display the minimum possible sum of disparities for partitioning the shipments into two groups.

Sample Input 1

```
5
4 5 0 2
1 3 7
2 0
4
```

Sample Output 1

```
4
```

Sample Input 2	Sample Output 2
7 1 10 5 5 5 5 5 10 5 5 5 100 100 5 5 10 5 5 98 99 3	15

Problem J
Oil 2

A large part of the world economy depends on oil, which is why research into new methods for finding and extracting oil is still active. Profits of oil companies depend in part on how efficiently they can drill for oil. The International Crude Petroleum Consortium (ICPC) hopes that extensive computer simulations will make it easier to determine how to drill oil wells in the best possible way.

Drilling oil wells optimally is getting harder each day – the newly discovered oil deposits often do not form a single body, but are split into many parts. The ICPC is currently concerned with stratified deposits, as illustrated in Figure J.1.

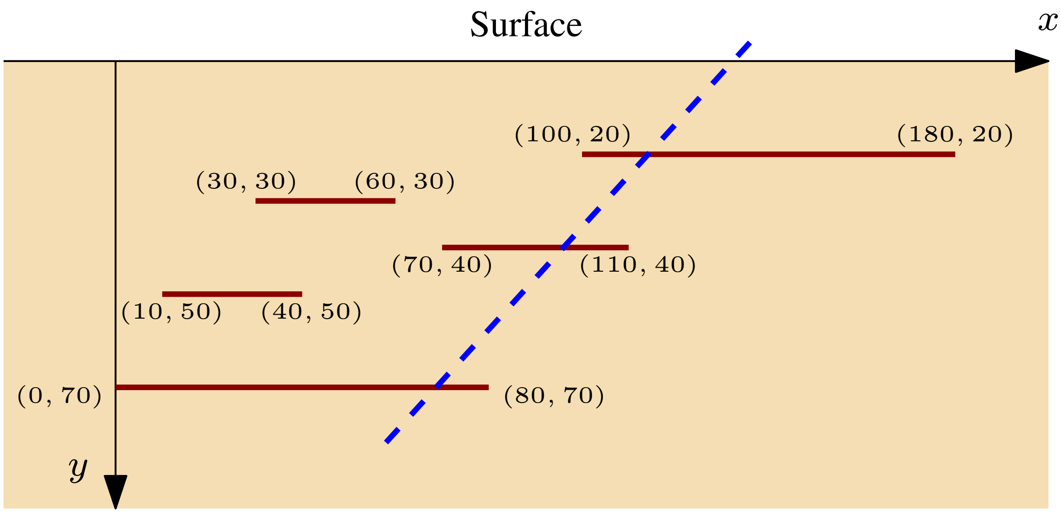


Figure J.1: Oil layers buried in the earth. This figure corresponds to Sample Input 1.

To simplify its analysis, the ICPC considers only the 2-dimensional case, where oil deposits are modeled as horizontal line segments parallel to the earth's surface. The ICPC wants to know how to place a single oil well to extract the maximum amount of oil. The oil well is drilled from the surface along a straight line and can extract oil from all deposits that it intersects on its way down, even if the intersection is at an endpoint of a deposit. One such well is shown as a dashed line in Figure J.1, hitting three deposits. In this simple model the amount of oil contained in a deposit is equal to the width of the deposit. Can you help the ICPC determine the maximum amount of oil that can be extracted by a single well?

Input

The first line of input contains a single integer n ($1 \le n \le 2\,000$), which is the number of oil deposits. This is followed by n lines, each describing a single deposit. These lines contain three integers x_0, x_1, and y giving the deposit's position as the line segment with endpoints (x_0, y) and (x_1, y). These numbers satisfy $|x_0|, |x_1| \le 10^6$ and $1 \le y \le 10^6$. No two deposits will intersect, not even at a point.

Output

Display the maximum amount of oil that can be extracted by a single oil well.

Sample Input 1

```
5
100 180 20
30 60 30
70 110 40
10 40 50
0 80 70
```

Sample Output 1

```
200
```

Sample Input 2

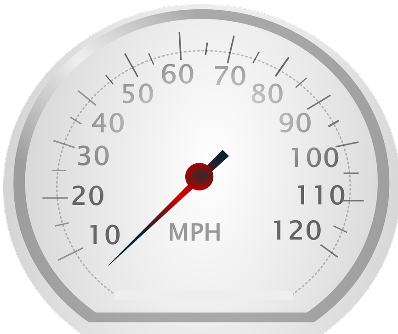
```
3
50 60 10
-42 -42 20
25 0 10
```

Sample Output 2

```
25
```

Problem K
Need for Speed 2

Sheila is a student and she drives a typical student car: it is old, slow, rusty, and falling apart. Recently, the needle on the speedometer fell off. She glued it back on, but she might have placed it at the wrong angle. Thus, when the speedometer reads s, her true speed is $s + c$, where c is an unknown constant (possibly negative).



Sheila made a careful record of a recent journey and wants to use this to compute c. The journey consisted of n segments. In the i^{th} segment she traveled a distance of d_i and the speedometer read s_i for the entire segment. This whole journey took time t. Help Sheila by computing c.

Note that while Sheila's speedometer might have negative readings, her true speed was greater than zero for each segment of the journey.

Input

The first line of input contains two integers n ($1 \leq n \leq 1\,000$), the number of sections in Sheila's journey, and t ($1 \leq t \leq 10^6$), the total time. This is followed by n lines, each describing one segment of Sheila's journey. The i^{th} of these lines contains two integers d_i ($1 \leq d_i \leq 1\,000$) and s_i ($|s_i| \leq 1\,000$), the distance and speedometer reading for the i^{th} segment of the journey. Time is specified in hours, distance in miles, and speed in miles per hour.

Output

Display the constant c in miles per hour. Your answer should have an absolute or relative error of less than 10^{-6}.

Sample Input 1

```
3 5
4 -1
4 0
10 3
```

Sample Output 1

```
3.000000000
```

Sample Input 2

```
4 10
5 3
2 2
3 6
3 1
```

Sample Output 2

```
-0.508653377
```

Problem L
Swap Space 2

You administer a large cluster of computers with hard drives that use various file system types to store data. You recently decided to unify the file systems to the same type. That is quite a challenge since all the drives are currently in use, all of them are filled with important data to the limits of their capacities, and you cannot afford to lose any of the data. Moreover, reformatting a drive to use a new file system may significantly change the drive's capacity. To make the reformat possible, you will have to buy an extra hard drive. Obviously, you want to save money by minimizing the size of such extra storage.

You can reformat the drives in any order. Prior to reformatting a drive, you must move all data from that drive to one or more other drives, splitting the data if necessary. After a drive is reformatted, you can immediately start using it to store data from other drives. It is not necessary to put all the data on the same drives they originally started on – in fact, this might even be impossible if some of the drives have smaller capacity with the new file system. It is also allowed for some data to end up on the extra drive.

As an example, suppose you have four drives A, B, C, and D with drive capacities 6, 1, 3, and 3 GB. Under the new file system, the capacities become 6, 7, 5, and 5 GB, respectively. If you buy only 1 GB of extra space, you can move the data from drive B there and then reformat drive B. Now you have 7 GB free on drive B, so you can move the 6 GB from drive A there and reformat drive A. Finally, you move the six total gigabytes from drives C and D to drive A, and reformat C and D.

Input

The input begins with a line containing one integer n ($1 \le n \le 10^6$), which is the number of drives in your cluster. Following this are n lines, each describing a drive as two integers a and b, where a is the capacity with the old file system and b is the capacity with the new file system.

All capacities are given in gigabytes and satisfy $1 \le a, b \le 10^9$. (One thousand petabytes should be enough for everyone, right?)

Output

Display the total extra capacity in gigabytes you must buy to reformat the drives.

Sample Input 1	Sample Output 1
4 6 6 1 7 3 5 3 5	1

Sample Input 2	Sample Output 2
4 2 2 3 3 5 1 5 10	5

This page is intentionally left blank.