J: You Win!

You just achieved the High Score on your favorite video game! Now, you get to enter your name! You have to use the controller to enter your name, which can be awkward. Here's how it works:

- There are only the 26 capital letters **A** to **Z**, in order. There are no numbers, spaces, lower case letters, or any other characters.
- Pushing UP or DOWN changes the active letter one letter forward (UP) or backward (DOWN). The active letter starts at **A**. It will not reset when you move around in the name. It also wraps: UP from **Z** goes to **A**, DOWN from **A** goes to **Z**.
- Pushing LEFT or RIGHT moves the cursor one letter left or right in the current name. Note that once the cursor is at either end of the current name, it cannot move any further in that direction.
- Pushing the FIRE button adds the active letter to the name.

For example, consider the name 'ALMA'. One way you could enter 'ALMA' is like this:

Action	# of Pushes	Name (| = Cursor)	Active Letter
FIRE	1	A|	A
UP	11	A|	L
FIRE	1	AL|	L
UP	1	AL|	M
FIRE	1	ALM|	M
DOWN	12	ALM|	A
FIRE	1	ALMA|	A

This would take 28 button pushes. However, consider entering 'ALMA' like this:

Action	# of Pushes	Name (| = Cursor)	Active Letter
FIRE	1	A|	A
FIRE	1	AA|	A
LEFT	1	A|A	A
UP	11	A|A	L
FIRE	1	AL|A	L
UP	1	AL|A	M
FIRE	1	ALM|A	M

This takes only 17 button pushes. Given a name, what is the fewest number of button pushes needed to enter that name? Assume that the active letter starts at **A**, and that it doesn't matter where the cursor ends up when you're done.

Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from 1 to 18 capital letters, representing a name that must be entered into the High Score list. The input will end with a line with a single **0**.

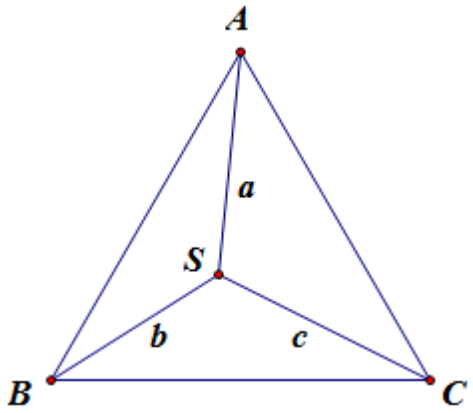
Output

For each test case, output a single integer representing the smallest number of button pushes needed to enter the name. Output no spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
ALMA YES 0	17 21

B: Stained Carpet

The Algebraist Carpet Manufacturing (ACM) group likes to produce area carpets based upon various geometric figures. The 2014 ACM carpets are all equilateral triangles. Unfortunately, due to a manufacturing defect, some of the carpets are not as stain-resistant as intended. The ACM group is offering to replace each defective carpet that contains a stain.



The web form used to report the stained carpet requests the three distances that the stain is away from the corners of the rug. Based upon these three numbers, you need to compute the area of the rug that is to be sent to the customer, or indicate that the customer's carpet doesn't come from ACM.

Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each test case will consist of a single line with three floating point numbers ***a***, ***b*** and ***c*** ($0<a,b,c\leq 100$) representing the distances from the stain to each of the three corners of the carpet. There will be a single space between ***a*** and ***b***, and between ***b*** and ***c***.

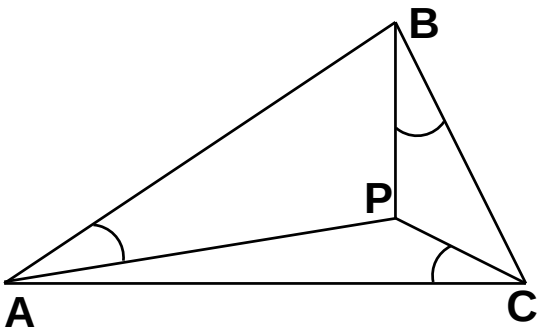
Output

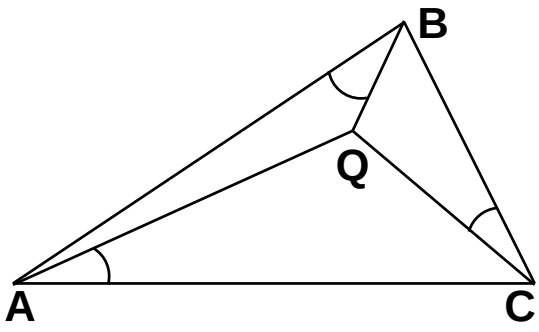
Output a single line with a single floating point number. If there is a carpet that satisfies the constraints, output the area of this carpet. If not, output `-1.000`. Output this number to exactly three decimal places, rounded. Output no spaces.

Sample Input	Sample Output
`1 1 1.732051`	`1.732`
`1 1 3.0`	`-1.000`
`1.732051 1.732051 1.732051`	`3.897`

D: Equal Angles

The All-Equal company has been tasked with placing towers on triangular plots so that the angles formed between the towers and the sides of the plots are equal. Given a triangle defined by points A, B and C, there are two such points - call them P and Q. There is one where angles PAB = PBC = PCA, and one where angles QBA = QCB = QAC.





Input

There will be several test cases in the input. Each test case will consist of six integers on a single line:

AX AY BX BY CX CY

Each integer will be in the range from -100 to 100. These integers represent the three points of the triangle: (**AX**,**AY**), (**BX**,**BY**) and (**CX**,**CY**). The points are guaranteed to form a triangle: they will be distinct, and will not all lie on the same line. The input will end with a line with six 0s.

Output

For each test case, output four space-separated real numbers:

PX PY QX QY

Where (**PX**,**PY**) and (**QX**,**QY**) are the requested points. Print each real number with exactly two decimal places, rounded, and put a single space between them. Print no blank lines between outputs.

Sample Input

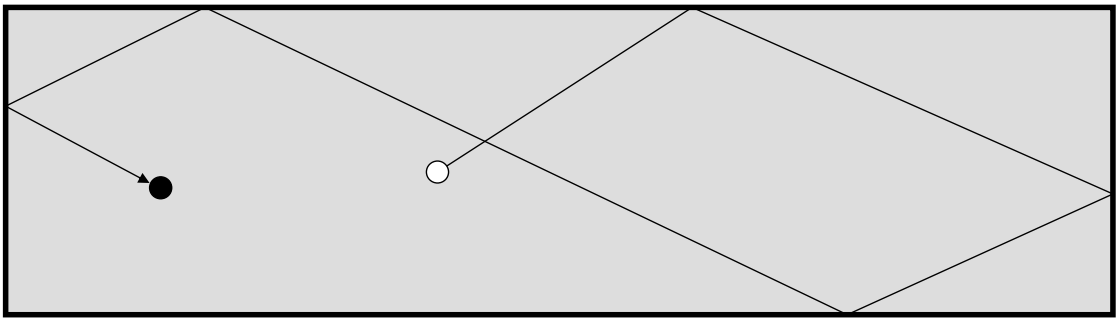
```
-4 5 -10 0 7 0
-15 -5 15 -5 0 21
0 0 0 0 0 0
```

Sample Output

```
-6.26 1.28 -1.96 3.07
0.01 3.66 -0.01 3.66
```

G: Pool Table

Consider a pool table with a cue ball and a target ball. The cue ball must bounce off of a certain number of cushions (i.e. edges of the table), and then hit the target ball. What is the minimum distance that the cue ball has to travel?



Assume ideal cushions (i.e., laws of reflection apply), and a negligible ball diameter. The coordinate system uses a corner of the table as the origin, and the edges of the table are aligned with the coordinate axes. If the cue ball hits in a corner, it is considered to be hitting two cushions. The cue ball must hit *exactly* the correct number of cushions first, *before* hitting the target the ball.

Input

There will be multiple test cases. Each case is on a single line containing seven integers:

L W CX CY TX TY N

The first two integers, **L** and **W** (**2 ≤ L,W ≤ 100**), are the dimensions of the table. The next two pairs of integers are the coordinates (**X**, **Y**) of the cue and target balls, such that **0** < **CX,TX** < **L**, and **0** < **CY,TY** < **W**, and (**CX**,**CY**) is not the same as (**TX**,**TY**). The last integer **N**, (**0 ≤ N ≤ 100**), is the number of cushions that must be hit. The test cases will be followed by a line with seven **0**'s.

Output

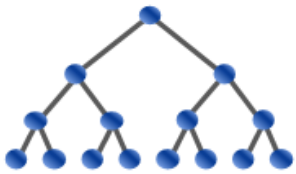
For each test case, print a single decimal number, rounded (NOT truncated) to 3 decimal places, representing the shortest distance the cue ball must travel. Print each answer on its own line, with no blank lines between answers.

Sample Input

```
20 15 10 1 12 1 1
10 20 1 2 7 16 2
0 0 0 0 0 0 0
```

Sample Output

```
2.828
19.698
```



USACO 2018 DECEMBER CONTEST, PLATINUM
PROBLEM 1. BALANCE BEAM

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

In order to save money for a new stall in her barn, Bessie the cow has started performing in the local circus, demonstrating her remarkable sense of balance as she carefully walks back and forth on an elevated balance beam!

The amount of money Bessie earns in her performance is related to where she manages to ultimately jump off the beam. The beam has positions labeled $0, 1, \ldots, N+1$ from left to right. If Bessie ever reaches 0 or $N+1$ she falls off one of the ends of the beam and sadly gets no payment.

If Bessie is at a given position k, she can do either of the following:

1. Flip a coin. If she sees tails, she goes to position $k-1$, and if she sees heads, she goes to position $k+1$ (i.e. $\frac{1}{2}$ probability of either occurrence).

2. Jump off the beam and receive payment of $f(k)$ $(0 \leq f(k) \leq 10^9)$.

Bessie realizes that she may not be able to guarantee any particular payment outcome, since her movement is governed by random coin flips. However, based on the location where she starts, she wants to determine what her expected payment will be if she makes an optimal sequence of decisions ("optimal" meaning that the decisions lead to the highest possible expected payment). For example, if her strategy earns her payment of 10 with probability $1/2$, 8 with probability $1/4$, or 0 with probability $1/4$, then her expected payment will be the weighted average $10(1/2) + 8(1/4) + 0(1/4) = 7$.

INPUT FORMAT (file balance.in):

The first line of input contains N $(2 \leq N \leq 10^5)$. Each of the remaining N lines contain $f(1) \ldots f(N)$.

OUTPUT FORMAT (file balance.out):

Output N lines. On line i, print out 10^5 times the expected value of payment if Bessie starts at position i and plays optimally, rounded down to the nearest integer.

SAMPLE INPUT:

```
2
1
3
```

SAMPLE OUTPUT:

```
150000
300000
```

Problem credits: Franklyn Wang and Spencer Compton

Contest has ended. No further submissions allowed.

Taxable Area

Filename: *area*
Time limit: *5 seconds*

Farmer Juan has a prize set of orange trees. Unfortunately, the county assess taxes on the property with those trees. The amount of the tax is proportional to the area that Farmer Juan claims as his own. Naturally, we wants to claim as little area as possible, while still including all of his trees. For the purposes of this problem, his trees may be treated as points in the Cartesian plane. The county has realized that if people are allowed to draw "crooked" (non-convex) property lines, then they can claim an exceedingly small area for their land while still having quite a few orange trees on it:

Thus, the county stipulates that all citizens must claim a single piece of land in the shape of a convex polygon. Thus, in the case shown above, Farmer Juan would be forced to claim the following area as his own:

(Note: This looks like a taco! Farmer Juan loves tacos!!!) Also, note that he's not allowed to split his trees into two separate convex polygons, as this could lead to some shenanigans as well. ***Note: If all of Farmer Juan's trees are collinear, he can claim 0 area!!!*** (These are the only shenanigans allowed.)

Since Farmer Juan is planning on changing getting new land every now and then, your program should process multiple sets of trees.

The Problem

Given a listing of where Farmer Juan's orange trees are, determine the minimum area he must claim so that he can claim all of his trees as his own. Essentially, find the area of the smallest convex polygon that contains all of his trees.

The Input

The first line of input will contain a single positive integer, s ($s \leq 100$), representing the number of sets of trees to evaluate. The input for each set of trees follows. The first line of each input set contains a single positive integer, n ($3 \leq n \leq 50000$), representing the number of orange trees for the input set. The following n lines each contain a pair of space-separated integers, x_i and y_i ($-10^4 \leq x_i , y_i \leq 10^4$), the x and y coordinates, respectively, of the i^{th} tree. Each of these n points is guaranteed to be distinct.

The Output

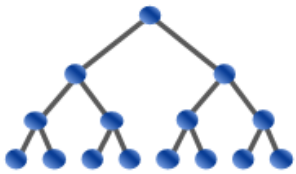
For each set of trees, output on a line by itself, twice the minimum area that Farmer Juan can claim while still claiming all of his orange trees. (Note: this value is always guaranteed to be an integer.)

Sample Input

```
2
7
0 0
1 1
2 1
3 2
5 5
5 0
0 5
4
3 4
7 7
5 6
4 9
```

Sample Output

```
50
17
```



USACO 2019 FEBRUARY CONTEST, PLATINUM
PROBLEM 1. COW DATING

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

Not impressed by the lackluster dating websites currently available to cows (e.g., eHarmoony, Moosk, Plenty of Cows), Farmer John decides to launch a new cow dating site based on a fancy proprietary matching algorithm that matches cows and bulls according to a wide range of their mutual interests.

Bessie, in searching for a partner to the Valentine's Day Barn Dance, has decided to try out this site. After making her account, FJ's algorithm has given her a list of N possible matches ($1 \leq N \leq 10^6$). Going through the list, Bessie concludes that each bull has probability p_i ($0 < p_i < 1$) of accepting an invitation from her for the dance.

Bessie decides to send an invitation to each bull in a contiguous interval of the list. Virtuous as always, she wants exactly one partner. Please help Bessie find the maximum probability of receiving exactly one accepted invitation, if she chooses the right interval.

INPUT FORMAT (file cowdate.in):

The first line of input contains N ($1 \leq N \leq 10^6$). Each of the remaining N lines contain 10^6 times p_i, which is an integer.

In at least 25% of the test cases, it is further guaranteed that $N \leq 4000$.

OUTPUT FORMAT (file cowdate.out):

Print 10^6 times the maximum probability of receiving exactly one accepted invitation, rounded down to the nearest integer.

SAMPLE INPUT:

```
3
300000
400000
350000
```

SAMPLE OUTPUT:

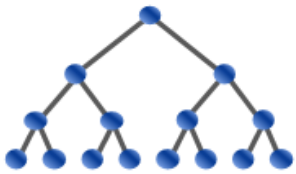
```
470000
```

The maximal probability results from selecting the interval from the 2nd to the 3rd cow.

As a note, you should be somewhat careful with floating point precision when solving this problem. We advise using at least "doubles" (64-bit floating-point numbers) and not "floats" (32-bit floating point numbers).

Problem credits: Ethan Guo

Contest has ended. No further submissions allowed.



USACO 2019 JANUARY CONTEST, PLATINUM
PROBLEM 1. REDISTRICTING

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

The cow mega-city Bovinopolis is redistricting! -- always a contentious political process between the two major cow breeds (Holsteins and Guernseys) living there, since both breeds want to make sure they retain sufficient influence in the Bovinopolis government.

The greater metropolitan area of Bovinopolis consists of a line of N pastures ($1 \leq N \leq 3 \cdot 10^5$), each containing a single cow, which is either a Holstein or a Guernsey.

The government of Bovinopolis wants to divide the greater metropolitan area into some number of contiguous districts, so that each district contains at most K pastures ($1 \leq K \leq N$), and every pasture is contained in exactly one district. Since the government is currently controlled by Holsteins, they want to find a way to redistrict which minimizes the number of Guernsey-majority or tied districts (a district is tied if the number of Guernseys equals the number of Holsteins).

A concerned coalition of Guernseys is trying to figure out how much damage might be done by the government's redistricting. Help them figure out the worst-case minimum number of districts which are either Guernsey-majority or tied.

INPUT FORMAT (file redistricting.in):

The first line contains a two space-separated integers N and K. The second line contains a string of length N. Each character is either 'H' or 'G', for Holstein or Guernsey.

OUTPUT FORMAT (file redistricting.out):

Please output the minimum possible number of districts that are Guernsey-majority or tied.

SAMPLE INPUT:

```
7 2
HGHGGHG
```

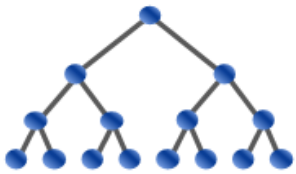
SAMPLE OUTPUT:

```
3
```

Problem credits: Dhruv Rohatgi

Contest has ended. No further submissions allowed.

USA Computing Olympiad



USACO 2018 DECEMBER CONTEST, PLATINUM
PROBLEM 2. SORT IT OUT

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

FJ has N ($1 \leq N \leq 10^5$) cows (distinctly identified $1 \ldots N$) lined up in a row. FJ likes his cows to be sorted in increasing order, but unfortunately they are currently out of order. While in the past FJ has used groundbreaking algorithms such as "bubble sort" to sort his cows, today he is feeling quite lazy. Instead he will yell at a specific cow, one at a time, to "sort it out". When yelled at, a cow will make sure she is not out of order (from her point of view). While there is a cow immediately to her right with a smaller ID, they will swap places. Then, while there is a cow immediately to her left with a larger ID, they will swap places. Finally, the cow is done "sorting it out", at which point the cow to her left will have a smaller ID and the cow to its right will have a larger ID.

FJ wants to pick a subset of cows, and then iterate through this subset, yelling at each of them in turn (in increasing order of ID), again and again until all N cows are sorted. For instance, if he picks the subset of cows with IDs $\{2, 4, 5\}$, then he will yell at cow 2, and then at cow 4, and then at cow 5. If the N cows are still not sorted, he will yell at these same cows again, and again, as necessary.

Since FJ is not sure which cows are paying attention, he wants to minimize the size of this subset. Furthermore, FJ thinks that the number K is very lucky. Help him find the K-th lexicographically smallest subset of minimal size so that shouting at them repeatedly will eventually result in all cows being sorted.

A subset S of $\{1, \ldots, N\}$ is said to be lexicographically smaller than a subset T if the list of elements in S (in increasing order) is lexicographically smaller than the list of elements in T (in increasing order). For instance, $\{1, 3, 6\}$ is lexicographically smaller than $\{1, 4, 5\}$.

Scoring: In cases worth $3/16$ of the points, $N \leq 6$ and $K = 1$. In additional cases worth $5/16$ of the points, $K = 1$. In additional cases worth $8/16$ of the points, no further constraints.

INPUT FORMAT (file itout.in):

The first line contains a single integer, N. The second line contains a single integer K ($1 \leq K \leq 10^{18}$). The third line contains N space-separated integers, representing the cows' numbers from left to right.

It is guaranteed that there will be at least K valid subsets.

OUTPUT FORMAT (file itout.out):

The first line of output should contain the size of the minimal subset. The remaining lines should contain the IDs of the cows in the K-th lexicographically smallest subset of minimal size, with one ID per line, listed in increasing order.

SAMPLE INPUT:

```
4 1
4 2 1 3
```

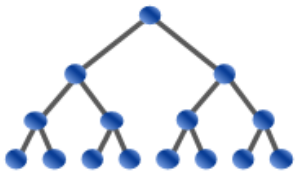
SAMPLE OUTPUT:

```
2
1
4
```

We start with the array `4 2 1 3`. After FJ yells at the cow with ID 1, the array will be `1 4 2 3`. When FJ yells at the cow with ID 4, the array will be `1 2 3 4`. At which point, the array is sorted.

Problem credits: Spencer Compton

Contest has ended. No further submissions allowed.



USACO 2018 JANUARY CONTEST, PLATINUM
PROBLEM 1. LIFEGUARDS

Return to Problem List

Contest has ended.

Log in to allow submissions in analysis mode

English (en)

Farmer John has opened a swimming pool for his cows, figuring it will help them relax and produce more milk.

To ensure safety, he hires N cows as lifeguards, each of which has a shift that covers some contiguous interval of time during the day. For simplicity, the pool is open from time 0 until time 10^9 on a daily basis, so each shift can be described by two integers, giving the time at which a cow starts and ends her shift. For example, a lifeguard starting at time $t = 4$ and ending at time $t = 7$ covers three units of time (note that the endpoints are "points" in time).

Unfortunately, Farmer John hired K more lifeguards than he has the funds to support. Given that he must fire exactly K lifeguards, what is the maximum amount of time that can still be covered by the shifts of the remaining lifeguards? An interval of time is covered if at least one lifeguard is present.

INPUT FORMAT (file lifeguards.in):

The first line of input contains N and K ($K \leq N \leq 100,000, 1 \leq K \leq 100$). Each of the next N lines describes a lifeguard in terms of two integers in the range $0 \ldots 10^9$, giving the starting and ending point of a lifeguard's shift. All such endpoints are distinct. Shifts of different lifeguards might overlap.

OUTPUT FORMAT (file lifeguards.out):

Please write a single number, giving the maximum amount of time that can still be covered if Farmer John fires K lifeguards.

SAMPLE INPUT:

```
3 2
1 8
7 15
2 14
```

SAMPLE OUTPUT:

```
12
```

In this example, FJ should fire the lifeguards covering $1 \ldots 8$ and $7 \ldots 15$.

Problem credits: Brian Dean

Contest has ended. No further submissions allowed.

Visiting Relatives

Filename: `relatives`

You must visit each of your relatives during the summer after you graduate college, because after that you start a job and have no idea when you'll have an extended amount of free time. Of course, since you don't have the job yet, you have little money and want to minimize your traveling costs.

The Problem:

Given the cost of traveling between each pair of locations, write a program to calculate the minimum cost to start at your home, travel to each relative exactly once, and return home. (Note: Thus, if you are at location 3 and want to go to location 5 next, and this cost is $45.99, but the cost of going from location 3 to 4 is $20.00 and the cost of going from 4 to 5 is $20,00, if you had previously visited location 4, you must still pay $45.99 for the direct route, since going via location 4 means visiting it twice, and you'd hate to show any one relative more love than the others, since they'd get jealous!)

The Input:

The first line of the input file will have a single positive integer, *T (T ≤ 20)*, representing the number of test cases in the file. The first line of each test case will contain a single positive integer*, n (n ≤ 15)*, representing the total number of locations you must visit, including your own home. The data for the case follows on the following *n* lines. The first of these *n* lines will contain the cost of traveling from your home (location 0) to all locations, numbered 0 through *n-1*, respectively. All of these costs will be positive real numbers expressed to exactly 2 decimal places, less than 1000. The following *n-1* lines will contain the corresponding traveling costs from locations 1 through *n-1*, respectively.

The Output:

For each test case, output the cost in dollars, rounded to two decimal places for the minimal traveling cost of starting from your home, visiting each relative exactly once and returning home.

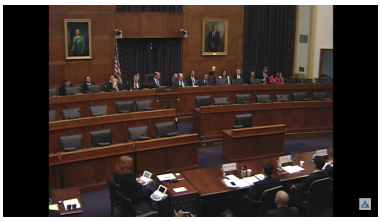
Sample Input:

```
1
3
0.00 2.00 4.00
3.00 0.00 5.00
2.50 5.50 0.00
```

Sample Output:

```
9.50
```

Problem I
The Uncertainty of Politics



You have an upcoming trip to Washington D.C. and you are fascinated with the intricacies of Congressional committee hearings. You wish to attend as many hearings as possible during your trip, and your local representative has provided you with a pass that will get you into the audience of any hearing. But there are some challenges in planning your schedule. Specifically:

1. There are many committees, and thus many hearings, some of which take place at overlapping times.
2. While the committees are extremely punctual in terms of when to start a hearing, they are notoriously unpredictable in terms of how long the hearing lasts. Fortunately, rules do not actually allow for a filibuster of a committee hearing, so they cannot last forever.
3. It is considered rude to enter a hearing that is already underway, or to leave a hearing before it is complete. Given that you do not wish to embarrass the representative who provided your tickets, if you attend you must attend the entire hearing. Fortunately, hearings are very near to each other; as soon as one hearing is done, you can immediately join another hearing that is about to start.

Well in advance of your trip, Congress publishes a schedule of hearings, indicating for each one the time s at which the hearing will start, and then values a and b which represent, respectively, the shortest and longest possible length of that particular hearing. You are to assume that the actual length of the hearing will be a uniformly random *integer* over the inclusive interval $[a, b]$.

Your goal is to develop a strategy that maximizes the expected number of hearings that you can attend during your trip. As an example, consider a situation in which there are four hearings with parameters as follows:

hearing	s	a	b
Social media and elections	1	1	7
NASA missions	3	2	3
Oil and gas exploration	5	1	4
Hurricane recovery efforts	6	10	10

For this schedule, the optimal strategy will allow you to achieve an expected value of 2.125 hearings. To achieve this, you begin by attending the NASA hearing, which starts at time 3 and ends with equal probability at either time 5 or time 6 (given the hearing length that is uniformly distributed over $\{2, 3\}$). If the NASA hearing does end at time 5 you will immediately head to the oil and gas exploration hearing, and there is a $\frac{1}{4}$ chance that hearing will end at time 6, allowing you to make yet a third hearing (about hurricane recovery efforts). If the NASA hearing instead ends at time 6, you will go straight to the hurricane hearing.

By this strategy you will attend 3 hearings 12.5% of the time and 2 hearings the other 87.5% of the time, and thus expected value of 2.125. Note that if you were to start by attending the social media and elections hearing, you might optimistically make four hearings. However, a careful analysis will demonstrate that if you attend the first hearing, your optimal expected value is only 2.10714.

Input

The input begins with an integer n that designates the total number of scheduled hearings ($1 \leq n \leq 10^4$). Following that are n lines, each containing three integers s, a, and b, respectively representing the start time, minimum length, and maximum length of a hearing, such that $1 \leq s \leq 10^6$ and $1 \leq a \leq b \leq 10^6$. The hearings will be listed in nondecreasing order of their start times.

Output

Display the expected number of hearings of an optimal strategy. Your answer should have an absolute or relative error of at most 10^{-3}.

Sample Input 1	Sample Output 1
4 1 1 7 3 2 3 5 1 4 6 10 10	2.125

Sample Input 2	Sample Output 2
5 1 1 7 1 1 6 3 2 3 5 1 4 6 10 10	2.29166667