A. Robo-Footballer

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

It's a beautiful April day and Wallace is playing football with his friends. But his friends do not know that Wallace actually stayed home with Gromit and sent them his robotic self instead. Robo-Wallace has several advantages over the other guys. For example, he can hit the ball directly to the specified point. And yet, the notion of a giveaway is foreign to him. The combination of these features makes the Robo-Wallace the perfect footballer — as soon as the ball gets to him, he can just aim and hit the goal. He followed this tactics in the first half of the match, but he hit the goal rarely. The opposing team has a very good goalkeeper who catches most of the balls that fly directly into the goal. But Robo-Wallace is a quick thinker, he realized that he can cheat the goalkeeper. After all, they are playing in a football box with solid walls. Robo-Wallace can kick the ball to the other side, then the goalkeeper will not try to catch the ball. Then, if the ball bounces off the wall and flies into the goal, the goal will at last be scored.

Your task is to help Robo-Wallace to detect a spot on the wall of the football box, to which the robot should kick the ball, so that the ball bounces once and only once off this wall and goes straight to the goal. In the first half of the match Robo-Wallace got a ball in the head and was severely hit. As a result, some of the schemes have been damaged. Because of the damage, Robo-Wallace can only aim to his right wall (Robo-Wallace is standing with his face to the opposing team's goal).

The football box is rectangular. Let's introduce a two-dimensional coordinate system so that point $(0, 0)$ lies in the lower left corner of the field, if you look at the box above. Robo-Wallace is playing for the team, whose goal is to the right. It is an improvised football field, so the gate of Robo-Wallace's rivals may be not in the middle of the left wall.

In the given coordinate system you are given:

- y_1, y_2 — the y-coordinates of the side pillars of the goalposts of robo-Wallace's opponents;
- y_w — the y-coordinate of the wall to which Robo-Wallace is aiming;
- x_b, y_b — the coordinates of the ball's position when it is hit;
- r — the radius of the ball.

A goal is scored when the center of the ball crosses the OY axis in the given coordinate system between $(0, y_1)$ and $(0, y_2)$. The ball moves along a straight line. The ball's hit on the wall is perfectly elastic (the ball does not shrink from the hit), the angle of incidence equals the angle of reflection. If the ball bounces off the wall not to the goal, that is, if it hits the other wall or the goal post, then the opposing team catches the ball and Robo-Wallace starts looking for miscalculation and gets dysfunctional. Such an outcome, if possible, should be avoided. We assume that the ball touches an object, if the distance from the center of the ball to the object is no greater than the ball radius r.

Input

The first and the single line contains integers y_1, y_2, y_w, x_b, y_b, r ($1 \le y_1, y_2, y_w, x_b, y_b \le 10^6$; $y_1 < y_2 < y_w$; $y_b + r < y_w$; $2 \cdot r < y_2 - y_1$).

It is guaranteed that the ball is positioned correctly in the field, doesn't cross any wall, doesn't touch the wall that Robo-Wallace is aiming at. The goal posts can't be located in the field corners.

Output

If Robo-Wallace can't score a goal in the described manner, print "-1" (without the quotes). Otherwise, print a single number x_w — the abscissa of his point of aiming.

If there are multiple points of aiming, print the abscissa of any of them. When checking the correctness of the answer, all comparisons are made with the permissible absolute error, equal to 10^{-8}.

It is recommended to print as many characters after the decimal point as possible.

Examples

input

```
4 10 13 10 3 1
```

output

```
4.3750000000
```

input

```
1 4 6 2 2 1
```

output

```
-1
```

input

```
3 10 15 17 9 2
```

output

```
11.3333333333
```

Note

Note that in the first and third samples other correct values of abscissa x_w are also possible.

B. Sweets for Everyone!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For he knew every Who down in Whoville beneath, Was busy now, hanging a mistletoe wreath. "And they're hanging their stockings!" he snarled with a sneer, "Tomorrow is Christmas! It's practically here!"

Dr. Suess, How The Grinch Stole Christmas

Christmas celebrations are coming to Whoville. Cindy Lou Who and her parents Lou Lou Who and Betty Lou Who decided to give sweets to all people in their street. They decided to give the residents of each house on the street, one kilogram of sweets. So they need as many kilos of sweets as there are homes on their street.

The street, where the Lou Who family lives can be represented as n consecutive sections of equal length. You can go from any section to a neighbouring one in one unit of time. Each of the sections is one of three types: an empty piece of land, a house or a shop. Cindy Lou and her family can buy sweets in a shop, but no more than one kilogram of sweets in one shop (the vendors care about the residents of Whoville not to overeat on sweets).

After the Lou Who family leave their home, they will be on the first section of the road. To get to this section of the road, they also require one unit of time. We can assume that Cindy and her mom and dad can carry an unlimited number of kilograms of sweets. Every time they are on a house section, they can give a kilogram of sweets to the inhabitants of the house, or they can simply move to another section. If the family have already given sweets to the residents of a house, they can't do it again. Similarly, if they are on the shop section, they can either buy a kilo of sweets in it or skip this shop. If they've bought a kilo of sweets in a shop, the seller of the shop remembered them and the won't sell them a single candy if they come again. The time to buy and give sweets can be neglected. The Lou Whos do not want the people of any house to remain without food.

The Lou Whos want to spend no more than t time units of time to give out sweets, as they really want to have enough time to prepare for the Christmas celebration. In order to have time to give all the sweets, they may have to initially bring additional k kilos of sweets.

Cindy Lou wants to know the minimum number of k kilos of sweets they need to take with them, to have time to give sweets to the residents of each house in their street.

Your task is to write a program that will determine the minimum possible value of k.

Input

The first line of the input contains two space-separated integers n and t ($2 \le n \le 5 \cdot 10^5$, $1 \le t \le 10^9$). The second line of the input contains n characters, the i-th of them equals "H" (if the i-th segment contains a house), "S" (if the i-th segment contains a shop) or "." (if the i-th segment doesn't contain a house or a shop).

It is guaranteed that there is at least one segment with a house.

Output

If there isn't a single value of k that makes it possible to give sweets to everybody in at most t units of time, print in a single line "-1" (without the quotes). Otherwise, print on a single line the minimum possible value of k.

Examples

input

```
6 6
HSHSHS
```

output

```
1
```

input

```
14 100
...HHHSSS...SH
```

output

```
0
```

input

```
23 50
HHSS.......SSHHHHHHHHHH
```

output

```
8
```

Note

In the first example, there are as many stores, as houses. If the family do not take a single kilo of sweets from home, in order to treat the inhabitants of the first house, they will need to make at least one step back, and they have absolutely no time for it. If they take one kilogram of sweets, they won't need to go back.

In the second example, the number of shops is equal to the number of houses and plenty of time. Available at all stores passing out candy in one direction and give them when passing in the opposite direction.

In the third example, the shops on the street are fewer than houses. The Lou Whos have to take the missing number of kilograms of sweets with them from home.

C. Piglet's Birthday

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Piglet has got a birthday today. His friend Winnie the Pooh wants to make the best present for him — a honey pot. Of course Winnie realizes that he won't manage to get the full pot to Piglet. In fact, he is likely to eat all the honey from the pot. And as soon as Winnie planned a snack on is way, the pot should initially have as much honey as possible.

The day before Winnie the Pooh replenished his honey stocks. Winnie-the-Pooh has n shelves at home, each shelf contains some, perhaps zero number of honey pots. During the day Winnie came to the honey shelves q times; on the i-th time he came to some shelf u_i, took from it some pots k_i, tasted the honey from each pot and put all those pots on some shelf v_i. As Winnie chose the pots, he followed his intuition. And that means that among all sets of k_i pots on shelf u_i, he equiprobably chooses one.

Now Winnie remembers all actions he performed with the honey pots. He wants to take to the party the pot he didn't try the day before. For that he must know the mathematical expectation of the number m of shelves that don't have a single **untasted pot**. To evaluate his chances better, Winnie-the-Pooh wants to know the value m after each action he performs.

Your task is to write a program that will find those values for him.

Input

The first line of the input contains a single number n ($1 \le n \le 10^5$) — the number of shelves at Winnie's place. The second line contains n integers a_i ($1 \le i \le n$, $0 \le a_i \le 100$) — the number of honey pots on a shelf number i.

The next line contains integer q ($1 \le q \le 10^5$) — the number of actions Winnie did the day before. Then follow q lines, the i-th of them describes an event that follows chronologically; the line contains three integers u_i, v_i and k_i ($1 \le u_i, v_i \le n$, $1 \le k_i \le 5$) — the number of the shelf from which Winnie took pots, the number of the shelf on which Winnie put the pots after he tasted each of them, and the number of the pots Winnie tasted, correspondingly.

Consider the shelves with pots numbered with integers from 1 to n. It is guaranteed that Winnie-the-Pooh Never tried taking more pots from the shelf than it has.

Output

For each Winnie's action print the value of the mathematical expectation m by the moment when this action is performed. The relative or absolute error of each value mustn't exceed 10^{-9}.

Examples

input

```
3
2 2 3
5
1 2 1
2 1 2
1 2 2
3 1 1
3 2 2
```

output

```
0.000000000000
0.333333333333
1.000000000000
1.000000000000
2.000000000000
```

D. Donkey and Stars

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the evenings Donkey would join Shrek to look at the stars. They would sit on a log, sipping tea and they would watch the starry sky. The sky hung above the roof, right behind the chimney. Shrek's stars were to the right of the chimney and the Donkey's stars were to the left. Most days the Donkey would just count the stars, so he knew that they are exactly n. This time he wanted a challenge. He imagined a coordinate system: he put the origin of the coordinates at the intersection of the roof and the chimney, directed the OX axis to the left along the roof and the OY axis — up along the chimney (see figure). The Donkey imagined two rays emanating from he origin of axes at angles α_1 and α_2 to the OX axis.

Now he chooses any star that lies strictly between these rays. After that he imagines more rays that emanate from this star at the same angles α_1 and α_2 to the OX axis and chooses another star that lies strictly between the new rays. He repeats the operation as long as there still are stars he can choose between the rays that emanate from a star.

As a result, the Donkey gets a chain of stars. He can consecutively get to each star if he acts by the given rules.

Your task is to find the maximum number of stars m that the Donkey's chain can contain.

Note that the chain must necessarily start in the point of the origin of the axes, that isn't taken into consideration while counting the number m of stars in the chain.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of stars. The second line contains simple fractions representing relationships "a/b c/d", such that $\frac{a}{b} = \frac{\sin\alpha_1}{\cos\alpha_1}$ and $\frac{c}{d} = \frac{\sin\alpha_2}{\cos\alpha_2}$ ($0 \le a, b, c, d \le 10^5$; $0^\circ \leqslant \alpha_1 < \alpha_2 \leqslant 90^\circ$; $\frac{a}{b} \ne \frac{0}{0}$; $\frac{c}{d} \ne \frac{0}{0}$). The given numbers a, b, c, d are integers.

Next n lines contain pairs of integers x_i, y_i ($1 \le x_i, y_i \le 10^5$)— the stars' coordinates.

It is guaranteed that all stars have distinct coordinates.

Output

In a single line print number m — the answer to the problem.

Examples

input

```
15
1/3 2/1
3 1
6 2
4 2
2 5
4 5
6 6
3 4
1 6
2 1
7 4
9 3
5 3
1 3
15 5
12 4
```

output

```
4
```

Note

In the sample the longest chain the Donkey can build consists of four stars. Note that the Donkey can't choose the stars that lie on the rays he imagines.

E. Endless Matrix

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A Russian space traveller Alisa Selezneva, like any other schoolgirl of the late 21 century, is interested in science. She has recently visited the MIT (Moscow Institute of Time), where its chairman and the co-inventor of the time machine academician Petrov told her about the construction of a time machine.

During the demonstration of the time machine performance Alisa noticed that the machine does not have high speed and the girl got interested in the reason for such disadvantage. As it turns out on closer examination, one of the problems that should be solved for the time machine isn't solved by an optimal algorithm. If you find a way to solve this problem optimally, the time machine will run faster and use less energy.

A task that none of the staff can solve optimally is as follows. There exists a matrix a, which is filled by the following rule:

The cells are consecutive positive integers, starting with one. Besides, $a_{i,j} < a_{t,k}$ ($i, j, t, k \geq 1$), if:

1. $max(i, j) < max(t, k)$;
2. $max(i, j) = max(t, k)$ and $j < k$;
3. $max(i, j) = max(t, k)$, $j = k$ and $i > t$.

So, after the first 36 numbers are inserted, matrix a will look as follows:

To solve the problem, you should learn to find rather quickly for the given values of x_1, y_1, x_2 and y_2 ($x_1 \leq x_2, y_1 \leq y_2$) the meaning of expression:

$$\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} a_{i,j}$$

As the meaning of this expression can be large enough, it is sufficient to know only the last 10 digits of the sought value.

So, no one in MTI can solve the given task. Alice was brave enough to use the time machine and travel the past to help you.

Your task is to write a program that uses the given values x_1, y_1, x_2 and y_2 finds the last 10 digits of the given expression.

Input

The first input line contains a single integer t ($1 \leq t \leq 10^5$) — the number of test sets for which you should solve the problem.

Each of the next t lines contains the description of a test — four positive integers x_1, y_1, x_2 and y_2 ($1 \leq x_1 \leq x_2 \leq 10^9, 1 \leq y_1 \leq y_2 \leq 10^9$), separated by spaces.

Output

For each query print the meaning of the expression if it contains at most 10 characters. Otherwise, print three characters "." (without the quotes), and then ten last digits of the time expression. Print the answer to each query on a single line. Follow the format, given in the sample as closely as possible.

Examples

input

```
5
1 1 1 1
2 2 3 3
2 3 5 6
100 87 288 2002
4 2 5 4
```

output

```
1
24
300
...5679392764
111
```