Problem A
Drawing Circles


"Colorful circle pattern" Vecteezy.com

Alice likes to draw circles. Whenever she has some time, she will grab a piece of paper and draw circles on it.

Alice is drawing circles on a paper with a coordinate system. Each circle is represented by an ordered tuple (x, y, r), where (x, y) is the position of the circle's center and r is the radius of the circle. Since she does not like two circles to cross each other, she always erases the inside area of each new circle she draws. That is, she always erases the interior of each new circle she draws, leaving the pencil line showing the circumference. (Of course, if she overwrites an existing circle then the new pencil line simply hides the old circumference, removing it from view.)

For example (see Figure A.1),

- if she first draws $(0, 0, 1)$ and then $(0, 0, 2)$, the first circle will be completely erased; or
- if she first draws $(-1, 0, 2)$ and then $(1, 0, 2)$, then the part of the first circle to the right of the $x = 0$ axis will be erased.

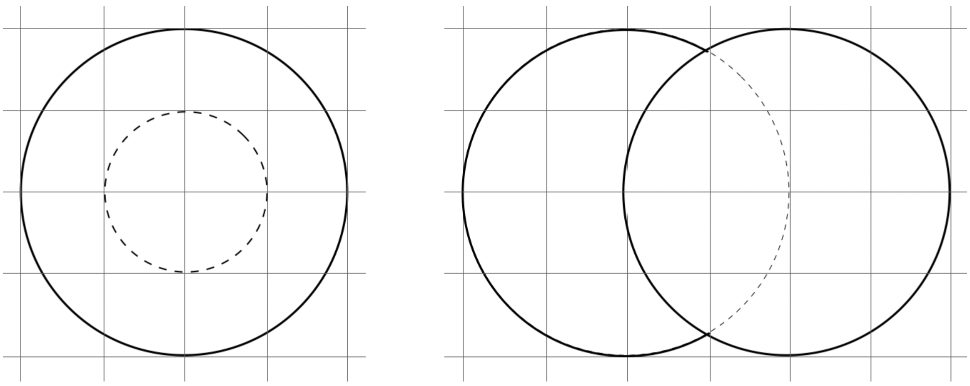


Figure A.1: Sample inputs. Dotted lines are areas that have been erased.

After drawing a number of circles, she would like to know the total length of all curved line segments that are still visible on the paper.

Input

The first line of input contains an integer N $(1 \leq N \leq 2000)$ denoting the number of circles Alice drew. Each of the following N lines contains three space-separated integers x_i, y_i, and r_i $(-10\,000 \leq x_i, y_i \leq 10\,000, 1 \leq r_i \leq 10\,000, i = 1 \ldots N)$ describing the i^{th} circle she drew.

Output

Output the total length of all curved line segments that are visible.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Sample Input 1	Sample Output 1
2 0 0 1 0 0 2	12.566371

Sample Input 2	Sample Output 2
2 -1 0 2 1 0 2	20.943951

Problem B
Perfect Path Patrol

Image by paulbr75 from Pixabay

Pleasantville is a community that appreciates simplicity. We can view the road network in Pleasantville as a collection of junctions that are connected by two-way streets. This has been done in a simple manner: there is precisely one way to travel between any two junctions in Pleasantville without traversing any street more than once.

Citizens have formed a community watch program to ensure the streets are safe to walk at night. So, some citizens patrol certain regions of the neighborhood. These patrols are also simple: a single citizen simply patrols all streets lying on the unique path between two junctions assigned to them.

Each street e also has a simple criterion: exactly p_e patrollers must include street e in their patrol path. If fewer than p_e patrollers were assigned to cover street e, then it might not be safe. If more than p_e patrollers were assigned to cover street e, the citizens themselves might become uneasy with the heightened presence of patrollers.

You have been tasked with organizing this community watch program. Of course, it is ideal to minimize the number of patrollers you use. Thus, you must enlist the fewest possible patrollers and assign each to a path between two junctions in the neighborhood such that any street e lies on exactly p_e patrollers' paths.

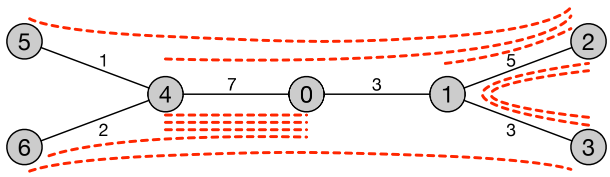


Figure B.1: An illustration of the first sample. The numbers by the solid black edges indicate how many patrollers must include that edge in their paths. The dashed red curves indicate one possible way to select 10 patrol paths so every edge lies on exactly the required number of patrol paths. That is, one solution is to use 10 patroller paths with endpoints:

$$(5, 2), (6, 0), (6, 3), (4, 2), (4, 0), (4, 0), (4, 0), (1, 2), (2, 3), (2, 3)$$

It is impossible to use fewer than 10 patrollers while ensuring each street is patrolled by exactly the required number of patrol paths.

Input

The first line of input contains a single value N ($2 \leq N \leq 500\,000$) indicating the number of junctions in the neighborhood, which are numbered 0 through $N - 1$.

Then $N - 1$ lines follow, each containing three integers u, v, p ($0 \leq u, v < N$, $0 \leq p \leq 10^9$). This indicates there is a street connecting junction u to junction v and that this street must lie on exactly p patrol paths.

You are guaranteed there is a unique way to travel between any two junctions using the provided streets.

Output

Output a single line with a single integer indicating the minimum number of patrollers you need to enlist for the community watch program.

Sample Input 1	Sample Output 1
7 0 1 3 1 2 5 1 3 3 0 4 7 4 5 1 4 6 2	10

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

Problem C
Sky's the Limit

"City with beautiful houses" designed by www.freepik.com

The N citizens of Eagleton have chosen to build their houses in a row along a single street (so that the houses can be numbered from 1 to N, with 1 the leftmost house and N the rightmost). The houses have varying heights, with house i being h_i inches tall.

Unfortunately, all is not well in Eagleton: the citizens have a bit of an envy problem. Every day, one random citizen (the owner of house i, let's say) emerges from their house and compares their house's height to the heights of the two neighboring houses. If house i is at least as tall as the average, plus k inches (in other words, if $h_i \geq (h_{i-1}+h_{i+1})/2+k$), the citizen retreats back into their house, satisfied. Otherwise, the citizen remodels their house to have new height $(h_{i-1}+h_{i+1})/2+k$. (The citizen does this remodeling even if the new height is only a tiny fraction of an inch taller than the old height—like we said, Eagleton has an envy problem.)

The left of house 1 and the right of house N is a nature preserve; the citizens of these houses treat the preserve as a "house" having height zero inches, for the purposes of the above calculations.

The city council of Eagleton is fed up with the constant construction traffic and noise, and has hired you to compute what Eagleton will look like when all of the remodeling is finally over. After some calculations, you discover that it is guaranteed that each house will converge to a final finite height after infinitely many days of the above remodeling process. Print the final height of the house that ends up tallest.

Input

The first line of input consists of an integer N and real number k, separated by a space ($1 \leq N \leq 100\,000; 0 \leq k \leq 10^{20}$): the number of houses in Eagleton and the number of inches each citizen wants their own house to be taller than the average of their neighbors.

N lines follow. The ith such line (starting at $i = 1$) contains a real number h_i ($0 \leq h_i \leq 10^{20}$), the initial height (in inches) of house i.

The real numbers in the input may be provided in scientific notation (see for instance Sample Input 2) and will have at most ten digits after the decimal point.

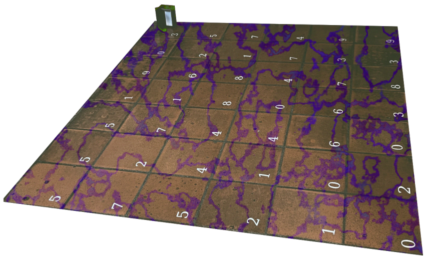
Output

Print the height (in inches) of the tallest house, after all houses have been remodeled over infinitely many days of the process described above. Your answer will be judged correct if the absolute or relative error is within 10^{-6}.

Sample Input 1	Sample Output 1
3 1 39 10 40	40.5

Sample Input 2	Sample Output 2
5 0.1 1.01e6 1.0e3 100 20.45 0	1010000

Problem D
Using Digits



From Gorman Lobby at University of Dallas

As the door slams shut behind Agent Youdy, he realizes that his daring escape from his arch-enemy's lair is not quite done. Before him lies a grid of numbered squares, and off in the distance he sees the exit door. He recognizes this setup from his training. It's a Sum Trap, and works as follows:

He must begin by stepping onto the square closest to himself, which we can think of as square $(1, 1)$, and walk to some other square, which we can think of as square (X, Y) for some $X, Y > 1$. Each step he takes is in either the positive-X or positive-Y direction, one square at a time. Every square has a number on it in the range 0 to 9, and the grid carefully monitors which squares he steps on. When he reaches the door, if the sum of the squares that he stepped on is sufficiently small, the door will open and he will be free. If not, he will be trapped forever.

After scanning the board for a while Agent Youdy becomes convinced that there is no path to success. But then he remembers a cryptic Code Key that he found while searching the lair. He reaches into his pocket and removes a paper with a large integer on it, each of whose digits is in the range 1 to 9. His training kicks in again, and he remembers the way this works: whenever he wishes, he may use the leading digit of the key, which permits him to hop over exactly that many squares. Whenever he uses a leading digit, it is removed from the number. Note that hops, no matter how far, change only his x- or y-coordinate, but not both.

For example, starting at the "1" on the bottom left corner in the figure shown, with Code Key 11, his best path is step right, step right, hop right, hop up, step right, step up, for a total of $1+0+3+2+1+1+8 = 16$. But starting with Code Key 12, his best path is step right, hop up, hop right, step right, step up, for a total of 18.

3	2	9	1	7	8
9	7	8	2	1	1
8	3	7	6	9	8
1	0	3	7	2	5

Figure D.1: Illustration for the sample inputs.

Input

The first line of input contains two space-separated integers X and Y, giving the width and height of the grid ($1 \leq X, Y \leq 100$). The next line contains a positive integer, the Code Key, guaranteed to be less than 10^{50} and containing only positive digits. Then follow Y lines, each containing X digits, giving the numbers on the grid of squares. The last digit of the first of these lines represents Agent Youdy's destination, and the first digit of the last line represents his starting square.

Output

The smallest sum possible on a path from $(1, 1)$ to (X, Y), including the endpoints.

Sample Input 1	Sample Output 1
6 4 11 329178 978211 837698 103725	16

Sample Input 2	Sample Output 2
6 4 12 329178 978211 837698 103725	18

Sample Input 3	Sample Output 3
6 4 23232345 329178 978211 837698 103725	11