

Problem A: Find My Arup

Filename: arup

Time limit: 2 seconds

As everyone has an Arup, a brilliant idea for the next killer app is obviously one that can find your Arup. But how can you ensure that you will find your Arup and not someone else's Arup. Your Arup is special and you cannot settle for finding just any old Arup.

That is why the find my Arup app will feature special advanced *Arup Locating Technology*TM. To help find your Arup, other Arup's will be used as *Telepathic Arup Beacons*TM. Specifically, each Arup i , not your Arup, is currently in the position (x_i, y_i) . Your Arup is in some position (x_a, y_a) . Each Arup, not your Arup, will send the result of the following function to your phone:

$$d(i) = \max(|x_i - x_a|, |y_i - y_a|)$$

From this information, it should be possible to locate your Arup precisely. If not, either two things have happened, there is not enough information to uniquely locate your Arup, or there is an issue with the *Arup Locating Technology*TM and your Arup's location is not consistent with the information given. You should determine which is the case and print Arup's location if possible. (Note: Your app should also take into account that all Arups, even your Arup, will always stand at integer coordinates and no two Arups, even your Arup, will occupy the same position.)

Input

The first line of input will be a single integer n ($1 \leq n \leq 100$), representing the number of Arups, not including your Arup.

The next n lines contains three integers x_i , y_i , and $d(i)$ ($-100 \leq x_i, y_i \leq 100$, $0 \leq d(i) \leq 100$), representing the location of Arup i and his beacon function from above.

Output

If you can find your Arup, print his location as integer coordinates $x_a y_a$. If your Arup could be at multiple locations print "Multiple" without quotes. If a beacon Arup must be misreporting, print "Error" without quotes.

Samples

Input	Output
1 1 1 0	Error
1 1 1 1	Multiple
2 -1 -1 1 2 2 2	0 0

Problem B: Quadruple Chomp

Filename: chomp4

Time limit: 3 seconds

Nom Chomsky the chain chomp regularly sneaks bites of food from Bowser's dinner table. All the food on the table is arranged on a line and Chomsky is able to eat any contiguous section in a single bite. After honing his technique he can impressively manage four bites before being caught. Chomsky knows how tasty each dish is wishes to maximize the combined tastiness of his four bites. However he must be careful as some dishes are so bad tasting that they have a negative tasty value! Find the maximum tastiness Chomsky can obtain with four or less bites.

Input

The first line of input contains a single positive integer, n ($n \leq 10^5$), representing the number of food items on the table. The second line contains n space-separated positive integers. The i^{th} integer, t_i ($-1000 \leq t_i \leq 1000$), on this line represents the tastiness of the i^{th} food item from the left on the table.

Output

For each test case output the maximum sum of tastiness Nom Chomsky can enjoy within four bites or less.

Samples

Input	Output
1 -1	0
2 1 -2	1
11 1 -9 5 -1 2 -3 1 -4 3 -4 1	11
9 2 -3 2 -3 3 -1 5 -5 3	14
9 10 -5 9 -100 30 -10 9 -6 10	63

Solution for the fourth sample test case with bites bolded: **2** -3 **2** -3 **3** -1 **5** -5 **3**

Solution for the fifth sample test case with bites bolded: **10** -5 **9** -100 **30** -10 **9** -6 **10**

Problem C: Garlic Bread Memes

Filename: garlicbread

Time limit: 3 seconds

While browsing a meme website, you come across the following post:

"You have been visited by the Great Garlic Bread! Upvote this post and some of the comments on this post in 3 seconds per test and luck shall rain down on you! But, you must upvote in a way that makes garlic bread look like it has a *long and increasingly positive* history. The comments are indexed from 1 to N in order of the time each comment was posted. Comment i has some garlic bread positivity rating $p(i)$, representing how much positive garlic bread emotions are expressed in comment i . Upvote this post and some *maximum-sized* subset of the comments S (sorted by indices) such that the slope between chosen comment S_i and S_{i+1} , defined as $\frac{p(S_{i+1})-p(S_i)}{S_{i+1}-S_i}$, is not less than K for the luck to rain down on you."

This post is likely just spam. In case you DO decide to upvote the post and the comments with the restrictions listed, what is the most number of comments you can upvote?

Input

The first line of input consists of two space separated integers, n ($1 \leq n \leq 1000$), the number of comments and k ($-10^9 \leq k \leq 10^9$), the minimum slope the subset of comments is allowed to have between adjacent comments. Line 2 describes the function p , where the i th integer equals $p(i)$ ($-10^9 \leq p(i) \leq 10^9$). All integers on both lines will be separated by spaces.

Output

Output the maximum number of comments you would upvote while following the restrictions of the post.

Samples

Input	Output
10 0 2 12 4 6 5 3 10 -5 8 10	5
10 -1 2 12 4 6 5 3 10 -5 8 10	7

Sample Input 1 Explanation: Upvote comments 1, 3, 4, 7, and 10. The slopes between consecutive comments in this sequence are 1, 2, 4/3, and 0, respectively, all greater than or equal to 0.

Sample Input 2 Explanation: Upvote comments 1, 3, 4, 5, 7, 9, and 10. The slopes between consecutive comments in this sequence are 1, 2, -1, 5/2, -1, and 2, respectively, all greater than or equal to -1.

Problem D: Universal Locker Rental 2

Filename: locker2

Time limit: 1 second

Universal Studios lets guests put their belongings in lockers while they enjoy thrill rides. As a courtesy, the lockers are free for up to one hour. After that, guests get charged a rather large amount of money for using a locker. Normally, you would put your belongings in a locker, enjoy a single ride and then collect your belongings. However, the process of checking out a locker and retrieving your items is rather laborious. Thus, it would be nice to minimize the number of times you have to check a locker. If the ride times are low enough, you could ride maybe two or three rides before having to retrieve your belongings and still not get charged at all.

Write a program that, given a list of the ride times (wait plus ride time) for all the rides you want to enjoy for a day, determines the minimum number of times you will need to check out a locker without paying any money. **You may select any order to ride the rides that minimizes this number.** For the purposes of this problem, assume that it takes $7\frac{1}{2}$ minutes to travel from a locker to any ride and $7\frac{1}{2}$ minutes to travel from any ride back to a locker.

Input

The first line of input contains a single positive integer, n ($1 \leq n \leq 40$), representing the number of rides you'd like to enjoy for the day. This is followed by n lines of input, each of which has a single positive integer, t ($15 \leq t \leq 45$), representing one of the ride times.

Output

On a single line, output the minimum number of times you will have to check out a locker to avoid paying for the locker at all.

Samples

Input	Output
5 15 20 35 30 20	3
4 15 20 25 35	3

For the first sample, the ordering (15, 30), (20, 20), (35), shows that three locker check outs suffice.

Problem E: Message Spam

Filename: spam

Time limit: 1 second

Somehow people keep on lending their phones to Zachary. The most logical thing to do in this situation is to go to the GroupMe CPSI 2017 unofficial chat from their phone and start spamming messages. When they get their phone back and see how they got roasted by the other campers, they will conclude that redemption is impossible and leave the group chat of their own volition.

Gabriel wants to add them back to group chat, but he needs know whose phones were in Zach's possession. He suspects that anyone who has submitted three or more identical messages is a victim. The messages need not necessarily be adjacent. For the purposes of this problem, when comparing two messages, do a **case sensitive** comparison. Thus, "Hi" and "hi" should NOT be treated as identical messages.

Input

The input begins with a single positive integer m ($m \leq 100$) the number of messages. Following this on each line will be a message of the form **<name>**: **<text>**. **<name>** will contain only alphabetic characters while **<text>** can contain alphabetic characters, numbers, spaces, and the symbols ! @ ? " , ' . (). **<name>** will have no more than 20 characters while **<text>** will have no more than 100. The first letter of **<name>** will always be uppercase. There will never be two or more spaces in a row or trailing spaces. **<name>** is always followed by a colon and a space (:).

Output

On separate lines, output the names of people in the chat who submitted the same message at least 3 times in alphabetical order by name. If there are no such people, output -1.

Samples

Input	Output
11 C: hey B: what's up A: the sky B: what's up A: the sky B: what's up A: the sky C: hey A: ho A: ho A: ho	A B

Input
3 Miles: anyone want to kill a giant roach?? lol please Ian: Call 911 Kurt: Evacuate the building
Output
-1

Input
22 Ethan: Hec 101 like normal? Right Ethan: ? Jacob: ? Chris: Im da map Chris: Im da map Chris: Im da map Chris: Im da map Chris: Im da map Kurt: ? Chris: Im da map Chris: Im da map Chris: Im da map Chris: Im da map Jeffrey: wait are you da map? Jeffrey: im not sure Jeffrey: you haven't made it clear enough Allen: burn the map smh Kian: Be proud of yourself chris Kian: You are the map Lina: I'm pretty sure "map" is an acronym for "most annoying person" Dylan: Rip Chris William: @Chris Pan might as well just leave the camp now
Output
Chris

Problem F: Superprime

Filename: superprime

Time limit: 1 second

Arup loves prime numbers. Not only that, but he loves prime numbers IN prime numbers. In a quest to get his name in the number theory books, he's come up with his own definition: a super prime number.

A super prime number is one such that each "prefix" of the number is prime as well. Consider the prime number 7393. Since each of its 4 prefixes, 7, 73, 739 and 7393 are prime, it is a super prime number!

Arup would like to enlist your help in studying super primes so that others may find them interesting and study them as well so that eventually he'll make a contribution to number theory books.

In particular, he'd like for you to write a program that takes in an integer k and returns the k^{th} super prime number, in numerical order.

Input

The input contains a single positive integer, k . The input will be such that it is guaranteed that the k^{th} super prime number exists.

Output

On a single line, output the k^{th} super prime number, in numerical order.

Samples

Input	Output
7	31
23	599

Note: The first seven super primes are 2, 3, 5, 7, 23, 29 and 31.

Problem G: Light Switch Combinations

Filename: lights

Time limit: 3 seconds

The normal room that Arup lectures in is inaccessible at the moment, so he settles for the room next door, which is rectangular and measures R by C meters. As it turns out, the electricians who wired the lights in this new room aren't all that great at what they do.

There are n light switches to this room and R times C lights in this room, one light for each square meter in the room. Each light switch may activate any subset of the R times C lights in this room. If two or more light switches that turn on the same light are flipped on, this light switch will stay on anyways.

So, what's the minimum amount of light switches Arup has to flip on in order to turn on all the lights in the room?

Input

The input begins with a line containing positive integers n , R , and C ($n \leq 13$, $R \cdot C \leq 30$). Following this are n descriptions of the light switches. Each one of these descriptions will consist of R lines and contain C characters. If the j^{th} character of the k^{th} line of the i^{th} light switch description is "#", that means turning on the k^{th} light switch turns on the light at row j , column i . Otherwise, this character is a ".".

Output

Output the minimum number of light switches necessary to turn on all the lights in the room, or "-1" if doing that is impossible.

Samples

Input 1	Output 1	Input 2	Output 2	Input 3	Output 3
3 2 3 ..# ##. .#. #.. ##. #.#	2	2 3 3 ..# ..# ..# #.. #.. #..	-1	6 1 6 #..... .#..... ..#.... ...#...#..#	6

On Sample 1: It is optimal to choose the first and third light switches.

On Sample 2: Even if every light switch is on, three of the lights won't activate.

On Sample 3: All switches need to be activated for every light to activate.

Problem H: Free the Rook

Filename: rook

Time limit: 5 seconds

Since Arup's kids were finicky the other morning, Matt (Garrison) had to teach the N-Queens problem. While Matt shares Arup's love of chess, his favorite piece is the rook, not the queen. Thus, Matt would like for you to solve a contrived problem about a rook, not queens! Here is the problem:

You are given a rectangular board with a single rook that appears to be trapped. There is a single square on the board that is marked as an exit, that is the rook's only path to freedom. Unfortunately, some of the squares on the board contain other pieces, which, for the purposes of this problem, the rook is not allowed to capture. Recall that a rook can move as many squares up, down, left or right on a chess board so long as there is no obstruction in the way. Write a program to determine the minimum number of moves the rook must make to reach the marked exit square.

Input

The first line of input has two space separated positive integers r ($r \leq 500$) and c ($c \leq 500$), representing the number of rows and columns, respectively, of the rectangular board. The following r lines contain c characters each, representing the characters on that particular row. Of these rc characters, exactly one will be 'R', marking the beginning location of the rook, exactly one will be 'E', marking the exit square, and the rest will either be '.', marking an empty square the rook can occupy or 'X', marking a square with another piece that the rook can't move to or through.

Output

On a single line, output the minimum number of moves necessary for the rook to move to the marked exit. If it's impossible for the rook to move towards the exit, output -1 instead.

Samples

Input 1	Output 1	Input 2	Output 2	Input 3	Output 3
4 6 RX.... .X.X.. ...X.. .X...E	4	6 5 X...R .X... E.X.. ..X.. .X...	-1	6 8 XRX...XE ..X.X.X. .X..X.X. ..X.X.X. X.X.X.X.X...	11

Problem I: Papa Smurf

Filename: smurf

Time limit: 1 second

At Universal, Ronald got a “kids meal” for dinner. His supposed excuse was that he wasn’t too hungry, but the real reason is that the “kids meal” came with a very valuable item, Papa Smurf:



It turns out the Papa Smurf is a very experienced programmer and Ronald has been using Papa since Monday to write Ronald’s programs for camp! As we all know, Papa is quite old. He learns from his mistakes. On average, if in one year he makes X number of mistakes per 1000 lines of code, he’ll cut out $X/2$ of those mistakes (using integer division). But, he’ll make 5 new mistakes since he becomes a bit more forgetful. For example, if Papa makes 27 mistakes per 1000 lines of code one year, he’ll cut out 13 (half of 27 using integer division) of those mistakes but add 5 new ones, so that he’ll make 19 mistakes per 1000 lines of code the following year. Write a program that calculates how many mistakes per 1000 lines Papa makes now.

Input

The input contains three space separated positive integers on a single line: a ($a \leq 10^4$), Papa’s age when he first started programming, m ($10 \leq m \leq 1000$), the number of mistakes per 1000 lines Papa made at age a , and b ($a \leq b \leq 10^{12}$), Papa’s current age. **Note: Papa might be really, really old now, even older than the universe!!!**

Output

On a single line, output the number of mistakes Papa makes per 1000 lines of code now.

Samples

Input	Output
25 27 26	19
14 1000 17	134