

Problem A. Ann Arbor

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Lots of bubble tea shops are opening in Ann Arbor! You are the owner of the most recently opened downtown bubble tea shop. Since it is too competitive for most tea shops to survive, you have decided to promote your brand by offering a free cup of bubble tea to every k -th customer on each day.

For example, if $k = 10$ and on day 1 you got 14 customers and on day 2 you got 33 customers, then you gave away 4 free cups: the 10th customer on day 1 and the 10th, 20th, and the 30th customers on day 2 each get one free cup.

You want to make sure that at least one customer gets a free drink each day. Write a program that given a log of visiting customers on the first D days, validates that on each day someone got a free drink, or, If not, outputs the first day that, sadly, no customer gets a free drink.

Input

The first line in each input file contains two integers k and D ($1 \leq k, D \leq 1000$). The next line contains D non-negative integers $a_1, a_2, \dots, a_D$, where a_i ($0 \leq a_i \leq 2000$) represents the number of visiting customers on day i .

Output

If on any day some customer gets a free drink, output **awesome**. Otherwise, output the smallest positive integer i , such that on day i no customer gets a free drink.

Examples

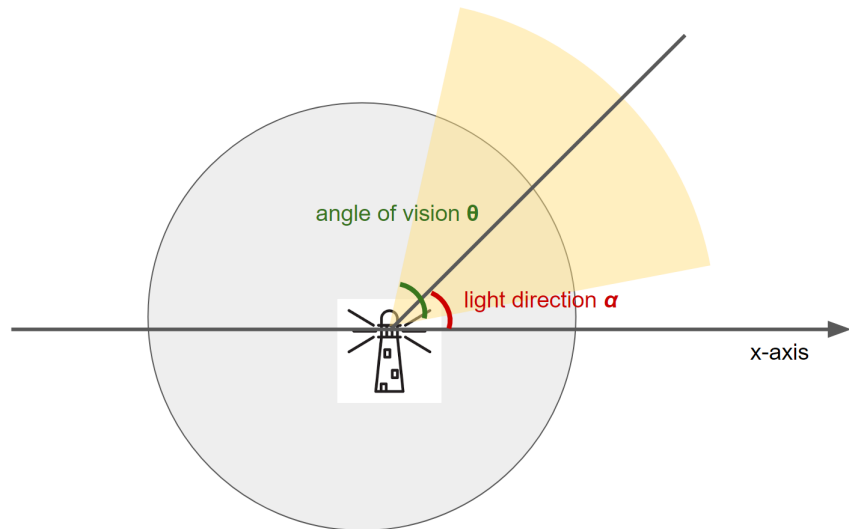
standard input	standard output
10 5 14 33 5 58 74	3
10 5 22 71 79 94 50	awesome
10 5 59 100 93 14 8	5

Problem B. Bois Blanc

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Bois Blanc is the French name for a special type of basswood. But if you are an expert on Michigan geography, you know that it is also a Michigan island at the northern end of Lake Huron. Of course, there are lots of basswood trees on the island. Unlike its famous and historical neighboring island – Mackinac Island – there are very few residents and visitors on Bois Blanc Island. It is known primarily for its lighthouses, which help guide ships entering and leaving the Mackinac Straits connecting Lake Huron and Lake Michigan.

Let us suppose that Bois Blanc is perfectly circular (it's not, but it's difficult to come up with a problem for the real Bois Blanc shape). Designate a Cartesian coordinate system where the origin $(0, 0)$ is at the center of the island and the island has radius 1. The State of Michigan has decided to build N state-of-the-art lighthouses on the island, numbered $1, 2, \dots, N$. Lighthouse i will be located at (x_i, y_i) and have a specific light angle α_i and an angle of vision θ_i (see figure). Thus, the lighthouse illuminates the area between two rays, the first extending from (x_i, y_i) at angle $\alpha_i - \theta_i/2$ and the second extending from (x_i, y_i) at angle $\alpha_i + \theta_i/2$.



You have just been appointed Michigan Lighthouse Commissioner (congratulations!). You would like to ensure that these lighthouses altogether light up the entire periphery of the island (if all the lights are on) so that ships do not run ashore on the island. Island periphery is the circumference of the island. You are tasked to write a program that certifies this for you.

Input

The first line in each input file contains an integer N ($1 \leq N \leq 100$), indicating the number of lighthouses on the island. In each of the next N lines, there are four real numbers $x_i, y_i, \alpha_i, \theta_i$ ($x_i^2 + y_i^2 < 1$; $0 \leq \alpha_i < 2\pi$; $0 < \theta_i < \pi$). Angles are given in radians.

The input is guaranteed that if you move the coordinates and angles of a lighthouse by $\pm 10^{-4}$, the answer will not change.

Output

For each test case, output **YES** if all lighthouses altogether light up the periphery of the island, and output **NO** otherwise.

Examples

standard input	standard output
4 0.2 0.2 0.785398 1.570796 -0.2 0.2 2.356194 1.570796 -0.2 -0.2 3.926990 1.570796 0.2 -0.2 5.497787 1.570796	NO
4 0.3 -0.2 0.785398 1.570796 0.2 0.3 2.356194 1.570796 -0.3 0.2 3.926990 1.570796 -0.2 -0.3 5.497787 1.570796	NO
4 0.2 -0.3 0.785398 1.570796 0.3 0.2 2.356194 1.570796 -0.2 0.3 3.926990 1.570796 -0.3 -0.2 5.497787 1.570796	YES
8 0.160778 -0.455910 2.270371 2.260020 -0.059877 -0.152423 2.615438 2.349560 0.092217 -0.908759 3.937581 2.103347 0.634763 0.439841 0.246451 0.232265 -0.291208 0.939505 6.245182 2.622931 0.005957 0.114427 1.133582 1.039842 -0.303909 0.652162 4.819728 0.999183 -0.625112 -0.292457 5.316244 0.593612	YES

Problem C. Canton

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Exotic Canton, home to shopping plazas and stores... many, many stores. Especially large furniture stores. You are a manager of one these large furniture stores (one that sells just furniture, no meatballs). It is time for you to design a new “in-store shopping route” for your customers. The entire store can be represented by an $N \times M$ grid of square blocks. Each block is a 20 foot by 20 foot square space featuring one special type of furniture. You have come up with several design plans. In each of the plans, there is a “suggested walking direction” in every block pointing to one of the four neighboring blocks. We will use characters ‘N’, ‘S’, ‘E’, and ‘W’ for each of the four directions north, south, east, west.

A customer entering your store is supposed to pick an arbitrary block, then start following the suggested directions from block to block. Unfortunately, it may sometimes happen that the customer is not able to exit the store by doing this. Please count the number of blocks such that a customer starting from that block and following the suggested walking directions will never exit the store.

Input

The first line in each input file contains two integers N and M ($1 \leq N, M \leq 2000$). In each of the next N lines there is a string of length M , composed of characters in ‘N’, ‘S’, ‘E’, ‘W’. As usual, the characters in each line are arranged from west to east and the lines are arranged from north to south.

Output

Please output the number of starting blocks from which it is not possible to exit the store.

Examples

standard input	standard output
3 3 SSW NSE NWS	7
3 4 SWNW EEEN NWWW	0
10 9 NENSSNWSN NSEEEENWE NWESSSEEE ESNNSEWNE WNEWENSWW SNNWESENS SNNWSSWNW ESNEEESSN NEWSSESWW NNWNENWWW	70

Problem D. Detroit

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

A cool thing about Detroit is that it is one of the few places where it is possible to go from the US to Canada by traveling south. You and K of your Detroit friends have made plans to meet in Detroit at the Ambassador Bridge and cross the border to Windsor, in Canada, to have a dim sum lunch. You will each be traveling from some location in Detroit to the Ambassador Bridge. The road layout in Detroit is very complicated so for this problem we will describe a (fictional) layout by specifying a directed graph with N vertices (or junctions) and M directed edges (or one-way roads). You and your K friends live at $K + 1$ different junctions in the city. The junctions are numbered from 1 to N . Junction 1 is the Ambassador Bridge, and you and your friends live at the junctions numbered from 2 to $K + 2$.

You and your friends want to use as few roads as possible. Therefore, your task is to design a transportation plan for you and your friends to travel to the Ambassador Bridge that minimizes the number of roads used. That is, you want to find the minimal number of edges you would need so that it is possible to go from each of the junctions $2, \dots, K + 2$ to junction 1 using just those edges.

Input

The first line of the input contains 3 integers: N , M , and K . ($3 \leq K + 2 \leq N \leq 20, N - 1 \leq M \leq 40$)

Then M lines follow: line i contains 2 integers u_i and v_i , indicating a one-way road from junction u_i to junction v_i .

Output

Output the minimum possible number of edges you and your friends need to travel in order to gather at junction 1. You may assume that there is at least one valid plan to do so.

Examples

standard input	standard output
5 5 2 2 1 3 2 4 3 3 5 5 1	3
7 7 2 2 1 6 2 7 6 3 7 4 3 3 5 5 1	4
7 6 1 2 1 3 1 4 1 5 1 6 1 7 1	2
7 7 2 1 2 2 3 3 4 4 1 2 5 6 2 6 7	3

Problem E. East Lansing

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

East Lansing is home to MSU, one of the Michigan Wolverines' greatest rivals (we dare not speak the name Ohio State). Every year at the Michigan-MSU football game, stadium bleachers fill with fans clad in either green t-shirts (one of the MSU colors), or blue t-shirts (one of the Michigan colors), or some other neutral color (why are these people here, anyway?). As you stand on the field staring up an $N \times M$ grid of seats, you notice that there are some connected regions of green clad fans, some connected regions of blue clad fans, and a few neutrally clad fans interspersed.

You see an opportunity to promote your newly acquired t-shirt company. You decide to "suggest" to some of the neutrally clad people that they commit: you will provide a green or blue t-shirt for them to wear, but you choose the color each person gets. They all agree to wear the shirt you provide because, after all, who doesn't want a free t-shirt? You decide to choose the colors in such a way that the number of connected regions (i.e., the number of green regions plus the number of blue regions) is minimized. You believe this will be a more pleasing arrangement in the advertising photo you plan to take. To clarify: if you can find a way to go from a seat to an adjacent seat in one of the four directions (up, down, right, or left) without changing the color, then these two seats are considered to be in the same connected region; you do not consider diagonally adjacent seats of the same color.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 100$). In each of the next N lines there is a string of length M consisting of characters '0', '1', and '?'. A '0' character means that the person in this seat is currently wearing green. A '1' character means that the person in this seat is currently wearing blue. A '?' character means that the person in this seat is wearing a neutral color. There will be at most 10 '?' characters in an input file.

Output

Output the minimum number of same color connected regions assuming you made the optimal suggestions (and they were accepted).

Examples

standard input	standard output
3 4 0101 ???? 0101	4
2 3 111 111	1
3 4 ???? ?01? ????	2
5 10 0000000011 0111?11111 0111001100 0?1111?000 110000000?	3

Problem F. Flint

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You were assuming Flint, Michigan. Fooled you! This problem is about a middle school teacher named Mr. Flint. One day, Mr. Flint created a new number theory game for his students. He prepared a set of N integers. Then, Mr. Flint asked the students, one-by-one, to choose a non-empty subset of those integers and compute the greatest common divisor of that subset. If the value is 1, then the subset is called a “comprehensive” subset and the student receives a point. Students may not choose a previously chosen subset.

To make sure that every student has an opportunity to receive a point, Mr. Flint comes to you, because you were his favorite student years ago and he has heard that you are good at computing. He asks you to compute the number of comprehensive subsets. He hopes this number is at least the size of the class.

Input

The first line contains an integer N ($1 \leq N \leq 100$). The next line contains the N positive integers $a_1, a_2, \dots, a_N$. ($1 \leq a_i \leq 10^9$) that Mr. Flint chose.

Output

Output the number of comprehensive subsets. Since this number may be large, output this number modulo $10^9 + 7$.

Examples

standard input	standard output
3 6 10 15	1
5 2 4 6 8 10	0
5 2 3 5 7 11	26

Problem G. Grand Rabbits

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 256 megabytes

Feeding rabbits is never an easy job, especially when the rabbits are large. In the city of Grand Rabbits there are N families of large rabbits that live along a long, straight street.

These families are numbered $1, 2, \dots, N$ as you traverse the street. You run a carrot delivery company that delivers carrots to the rabbit families each day. Whenever you deliver to the i -th family on the street, you leave an order of a_i pounds of carrots. You have worked out a schedule where you deliver on D consecutive days. On the d -th day of the schedule, you deliver to the families numbered from L_d to R_d , where $1 \leq L_d \leq R_d \leq N$. Also on the d -th day of your schedule, you have k_d trucks available to deliver these orders. To save wear on your trucks each day, you would like to apportion the $R_d - L_d + 1$ orders among the k_d trucks so that the maximum load is minimized. Furthermore, to save gas, each truck should deliver to a set of consecutively numbered families. For example, if there are 10 families in all, with $a_i = i$ for $i = 1, 2, \dots, 10$, and on a particular day d , $L_d = 1$, $R_d = 6$ and $k_d = 3$, then the first truck could deliver orders to families 1, 2, 3, the second truck to families 4,5, and the third truck to family 6. The second truck has the maximum load, 9 pounds, and this is minimal among all ways of apportioning the orders.

Input

In the first line there are 2 integers N and D ($1 \leq N, D \leq 10^5$), indicating the number of rabbit families and number of days your company runs. Then the next line contains N integers $a_1, a_2, \dots, a_N$ ($1 \leq a_i \leq 10^9$) indicating the order sizes of rabbit families. On each of the next D lines, there are 3 integers L_d, R_d, k_d ($1 \leq L_d \leq R_d \leq N, 1 \leq k_d \leq 10$), indicating the family range and the number of trucks on day d .

Output

For each day $d = 1, 2, \dots, D$, please output an integer indicating the maximum load of if you apportion orders optimally.

Example

standard input	standard output
10 4	9
1 2 3 4 5 6 7 8 9 10	19
1 6 3	15
3 10 3	10
1 10 5	
1 10 10	

Problem H. Holland

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

You are a manager of a beautiful but tiny take-out-only restaurant in Holland, a scenic city in western Michigan. All of your customers are regulars and you know that there will be N customers planning to order food from your restaurant today. You even know the arrival time of each customer, the food that customer will order, and the tip that customer will leave. (You have studied data science so you can predict these things perfectly!) The restaurant service works as follows. When a customer arrives at the front door, they will enter a waiting queue. Your restaurant always serves the customer at the front of the queue. After this customer gets their food they will leave the restaurant immediately and then you are able to serve the next customer. However, due to social distancing limitations these days, your restaurant can hold at most K customers at one time so you set this as the maximum queue size. Whenever a customer arrives at your restaurant and finds a full queue, that customer flies into a rage and leaves immediately. (If the queue is completely full with K customers and the customer at the front of the queue is served at exactly the same time that a $(K + 1)$ st customer arrives to join the queue, there is no problem: the served customer at the front will exit and the arriving customer will join the queue simultaneously – rage avoided.)

As an awesome manager, you would like to avoid the potential frustration from these customers. To do so, you would select a subset of customers and tell them not to order food today, so that all the remaining customers can be served. As a clever manager, you would like to maximize the total amount of tips you get from the remaining customers.

Input

The first line contains three integers N , K , and S ($1 \leq K \leq N \leq 1000, 1 \leq S \leq 10^6$), indicating the number of customers planning to order food from you today, the size of a queue, and the amount of time it takes to serve one customer. In each of the following N lines, there are two values a_i and t_i ($1 \leq a_i \leq 10^9, 1 \leq t_i \leq 10^6$), indicating the arrival time and the amount of tips you receive if they are served.

Output

Output the maximum possible amount of tips you can get.

Examples

standard input	standard output
3 2 10 1 100 6 200 8 300	500
3 2 10 1 100 6 200 12 100	400
3 1 10 1 100 6 200 17 100	300
10 3 10 1 120 4 105 8 134 11 104 13 114 26 111 17 113 16 126 19 111 25 129	623

Note

In the 4-th example, we should choose the customers that arrive at time 1, 8, 13, 16, and 25. In this way, the total tips we can get is $120 + 134 + 114 + 126 + 129 = 623$.

Problem I. Isle Royale

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

Isle Royale is the only national park in Michigan. Since Isle Royale is a large island located very far north in Lake Superior, people rarely visit during the winter, and the wildlife and natural scenery are quite well preserved. The only access to the island is by a ferry that runs once a day. You have decided to write a video game about an adventure on Isle Royale. The map of the island is modeled as a graph with N recreation sites (vertices) and M paths connecting these sites (undirected edges).

In this game, a hero (the main character) begins at the entrance, recreation site 1, where the ferry docks. Her objective is to travel along the paths to the campground at site N . Easy, right? But there are moose on the island! And a charging moose is three-quarters of a ton of dumb, terrifying fury. In fact, there is one moose at each recreation site i , where $1 \leq i \leq N - 1$.

Now there are two ways to deal with a moose (in the game – don't do this in real life): either stand perfectly still or climb a tree. The advantage of standing still is that it conserves energy, (but the moose does not leave). The advantage of climbing a tree is that the moose becomes frustrated and fades back into the forest, never to return, and the hero can advance. At each recreation site i , there is one tree and the hero must expend a certain amount of energy, P_i , to climb that tree. The hero begins the game with E energy units, which you may assume is positive. Whenever the hero is at a recreation site $i \leq N - 1$, she may spend one minute executing one of the three following actions:

- Stand still. If her energy when she arrives at i is less than E , she recovers one energy unit.
- Climb a tree. This costs her P_i energy units.
- Move along a path to another recreation site j . This option is available only when the moose has left. Also, the hero loses $D_{i,j}$ energy units walking along the path.

Our hero may never have a negative amount of energy, even when she reaches the destination site N . As the game designer, you would like to ensure that the game is difficult but fun. Can you write a program that calculates the shortest possible amount of time (in minutes) a hero needs to travel to site N ?

Input

The first line contains three integers N, M, E ($1 \leq N, M \leq 10^4$, $1 \leq E \leq 10^9$). In the second line there are $N - 1$ integers $P_1, P_2, \dots, P_{N-1}$ ($1 \leq P_i \leq E$). In each of the next M lines, there are three integers u_i, v_i , and D_{u_i, v_i} ($1 \leq u_i, v_i \leq N$, $0 \leq D_{u_i, v_i} \leq E$). You may assume that there is always a way for a hero to avoid the moose and get to the destination site N .

Output

Output the minimum amount of time it takes the hero to arrive at site N .

Examples

standard input	standard output
5 5 100 60 30 40 20 1 2 5 2 3 10 2 4 15 3 5 20 4 5 25	61
5 4 100 10 10 10 10 1 2 10 2 3 10 3 4 10 4 5 10	8
5 4 100 100 100 100 100 1 2 100 2 3 100 3 4 100 4 5 100	708

Problem J. Jackson

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Little Jackson is learning to spell! Recently he learned the word “michigan”. He is very excited to find all the occurrences of “michigan” on the street, even if they are spread out and hidden in other words. For example, ‘michigan’ occurs twice in ‘momichigan’ (because you can eliminate the first two letters, or the 2nd and 3rd letters). You can check that it occurs three times in ‘mimichiganmi’. One day, Little Jackson sees an electric newsboard on the street. There are N characters on this board, but due to bugs the screen sometimes goes blank just as he finishes his count. Little Jackson screams, “how can I check the michigans when the world is changing so fast?”

Take pity on Little Jackson. Write a program that checks Little Jackson’s answer before the screen goes blank.

Input

The first line contains a string S ($1 \leq |S| \leq 10^6$), consisting of characters ‘a’-‘z’ and underscore ‘_’ (which represents the whitespaces).

Output

Output a number that represents the occurrence of subsequences that equals to “michigan”. Since these numbers may be large, all you need to do is modulo the answer by $10^9 + 7$.

Examples

standard input	standard output
momichigan	2
mimichiganmi	3
magician	0