

Problem A

Archipelago

The company you work for, Boats to Get Out (BGO), has recently discovered a new archipelago that simply begs to become a new tourism hotspot. The islands are all extremely tiny, so there should be ample opportunity to base almost all transportation in the region on boats.



CC-BY 2.0, Tom Patterson via

[Flickr](#)

Unfortunately, the islands are so far out in the ocean that the standardized boat BGO provide is not able to reach any of them from the mainland; their boats can only hold enough fuel to travel a distance of d kilometers. To access the archipelago at all hence require that an airport is built on one of the islands. But where should it be located?

The BGO boss has ordered you to list out all the islands in order from highest to lowest *airport utility*. The airport utility of an island is defined as the number of islands it is possible to reach by boat from that island using any number of intermediate stops on other islands for refuelling.

Input

The first line of input consists of two space-separated integers $1 \leq n < 2\,000$ and $1 \leq d < 10^6$, where n signify the number of islands and d indicates how far a boat can travel in kilometers before it needs to refuel. The islands are named from 1 to n . The next n lines of input describes the location of the islands. The i^{th} such line contains two space-separated integers $0 \leq x_i < 10^7$ and $0 \leq y_i < 10^7$ which describe the coordinates of island i .

Output

Output a single line containing n space-separated integers indicating a ranking of the islands from highest to lowest airport utility. If there are multiple islands with the same utility, you may output them in any order as long as their airport utility is non-increasing.

Sample Input 1

```
7 3
1 1
3 2
2 3
4 2
12 5
13 7
11 6
```

Sample Output 1

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

Sample Input 2

```
3 4
2 5
5 0
4 4
```

Sample Output 2

```
1 3 2
```

CPU Time limit 2 seconds

Memory limit 1024 MB

Author Kristoffer Æsøy

Source Bergen Open 2021

License



Problem B

Bombs

Your team has discovered that the fearsome Bureau of Global Overlords (BGO) have devised a plan for world domination; the only way to save the world from certain doom is to blow up the BGO headquarters.



*Pixabay License,
OpenClipart-Vectors*

You have k bombs at your disposal, and an expert has analysed the headquarters super-intricate floor plan and pointed out the best rooms to place them. The last remaining problem is that the BGO headquarters has a somewhat peculiar surveillance system in their doors; the bombs are just a tiny tad below the threshold for the amount of suspiciousness the door surveillance accepts per day. Hence, each day only a single bomb may pass a given door in the headquarters. Moreover, the surveillance system is based on a type of ultrafancy wave technology that makes the bomb feel quite unwell – it is only safe to carry a particular bomb through a single door each day as well.

Assuming your stealth skills grants you unlimited access, how many days are required to place the bombs?

Input

The first line contains three space-separated integers, n , m and k . The first integer $1 \leq n \leq 100$ denotes the number of rooms in the BGO headquarters; the rooms are labelled with numbers from 1 to n (inclusive). For simplicity we denote the outside (where all the bombs are initially) as room 0.

The second integer $1 \leq m \leq 400$ denotes the number of doors between rooms in the headquarters. Note that there may be multiple doors between the same rooms, and that some doors may go to the outside. The third integer $1 \leq k \leq 8$ denotes the number of bombs.

On the second line follows k space-separated integers b_1, b_2, \dots, b_k , indicating the rooms where the bombs should be placed ($1 \leq b_i \leq n$ for every i).

Finally follows m lines, each describing a door. Each such line contains two distinct space-separated integers $0 \leq u, v \leq n$ indicating that there is a door between room u and room v .

Output

Output a single integer, the minimum number of days required to place the bombs.

Sample Input 1

```
3 4 5
2 2 2 3 3
0 1
1 2
2 3
2 0
```

Sample Output 1

```
3
```

CPU Time limit

3 seconds

Memory limit

1024 MB

Author

Torstein Strømme

Source

Bergen Open 2021

License



Problem C

Coins

You have been challenged by the infamous magician Zhivago to his signature coin game! The game starts off with a pile of n coins, where you and Zhivago will each take turns grabbing coins from the pile, and the loser is the one that grabs the last coin. Each player has to take 1, 2 or 3 coins each turn; no more, no less!



*CCo, via
Hippopx*

Zhivago is a master in his own game, and will always play optimally. However, you have been informed that you are in a winning position, and can best Zhivago to become the coin game champion. Can you do it?

Interaction

First, your program should read a single integer $1 \leq n < 1000$, the number of coins in the pile at the start of the game. Then, we repeat the following process:

- Your program writes a single line to the standard output, in the form of an integer 1, 2 or 3 that represents the move you make.
- After you have made your move, then your program should read a single integer $c \in \{1, 2, 3\}$ representing how many coins Zhivago takes on his turn.
- If there is only a single coin left for Zhivago, he will output the string “I give up” instead. At this point your program should terminate.

After making each move, do not forget to **flush** the output. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` or `print(output, flush=True)` in Python;
- see documentation for other languages.

Sample game

<i>Your output</i>	<i>Zhivago output</i>	<i>Interpretation</i>
	7	There are $n = 7$ coins as the starting amount in the pile.
2	1	You take 2 coins and Zhivago responds by taking 1 coin.
3	I give up	You take 3 coins. Since there is only 1 coin left, Zhivago gives up.

CPU Time limit

1 second

Memory limit

1024 MB

Author

Kristoffer Æsøy

Source

Bergen Open 2021

License



Problem D

Doomsday

Doomsday is near! Or at least that's what your brother is telling you. In his preparations he has constructed a clever network of well concealed food depots and water depots far out in a mountainous region. You are in your base, and the alarm goes off: how quickly can you fetch both food and water supplies?



*Public domain, Munds
Mountain Trail by Coconino
NF Photography via Flickr*

Input

The first line contains four integers n, m, w, f , where $1 \leq n \leq 50\,000$ is the number of hidden locations, $0 \leq m \leq 150\,000$ is the number of trails in the network, $1 \leq w \leq n$ is the number of water depots in total, and $1 \leq f \leq n$ is the number of food depots in total. Your base is at location 0. The second line contains w space-separated integers u_1, u_2, \dots, u_w , which represents the (distinct) locations of the water depots ($0 \leq u_i < n$ for each i). The third line contains f space-separated integers v_1, v_2, \dots, v_f , which represents the (distinct) locations of the food depots ($0 \leq v_i < n$ for each i).

The next m lines each describe a (bidirectional) trail in the network. The i^{th} such line contains three space-separated integers a_i, b_i and t_i indicating that there is a trail between location a_i and b_i which takes t_i hours to traverse ($0 \leq a_i, b_i < n$ and $0 \leq t_i < 100$ for each i).

Output

Output a single integer, the minimum number of hours required to fetch both food and water and bring it back to base.

Sample Input 1

```
7 7 2 2
3 6
4 5
0 1 3
0 2 1
1 3 3
1 4 1
2 5 2
2 6 10
4 5 1
```

Sample Output 1

```
14
```

CPU Time limit 5 seconds

Memory limit 1024 MB

Author Simen Hornnes

Source Bergen Open 2021

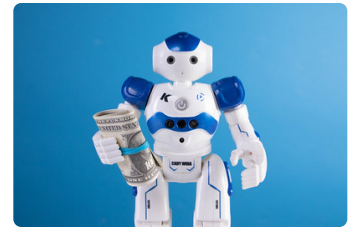
License



Problem E

Elder price robot

The idea behind your latest business adventure, BarGain Overview (BGO), is to collect the history of prices for a certain item that is available for sale on the web. The *BarGain score* of a particular day is defined as the number of days since the price was lower or equal to today's price. The worst BarGain score is thus 1, and if the price is strictly better than all previous prices ever recorded, then it is infinity. You want to report the BarGain score to your customers to help them identify a good buy.



CC-BY 2.0, Marco Verch via
Flickr

Input

The first line of input contains a single integer $2 \leq n \leq 400\,000$, the number of days for which you have collected price data for the item. On the second line of input follows n space-separated integers p_1, p_2, \dots, p_n , where $0 \leq p_i \leq 10^6$ is the price of the item $i - 1$ days ago. Today's price is p_1 .

Output

Output n rows, one for each day you have collected price data for the item. On the i^{th} such line, output the BarGain score for the day that was $i - 1$ days ago.

Sample Input 1

```
6  
9 10 11 9 12 11
```

Sample Output 1

```
3  
2  
1  
infinity  
1  
infinity
```

CPU Time limit 4 seconds

Memory limit 1024 MB

Author Torstein Strømme

Source Bergen Open 2021

License



Problem F

F1 Racing

All over Europe the F1-fever is burning stronger than ever after NextFlex released “Cars go vroooooom.” Once you get in, there is no way out. The drama, the tactics, the drama, the racing, but most of all the drama.

This year's series is one of the most exciting yet. The legendary battle between the two main stars Min Verstopen and Lery Slowington never fails to make a race entertaining. There are mind games, close battles and almost accidents all the way. But above all of this, the winner of a F1 race is the fastest driver, and the team with the best tactical decisions.



*CC-BY-2.0, Honda Pit Stop
by ariffjamili via Flickr*

An above average committed fan named Dorothy has noticed that in our modern era of F1 there are numerous instruments on a F1 car, giving the teams data on how fast each car can race in different conditions. Dorothy believes one could use superior knowledge of algorithms to decide the best possible tactic given a set of race conditions. She has run some statistics, and found that a car with x round old tires uses

$$r + b \cdot x$$

seconds on one lap where r and b are constants depending on the particular track. Now she turns to you to provide her with an algorithm that finds the total race time for a team using optimal pitstops.

Input

The input consists of four space-separated integers n , p , r and b .

- $1 \leq n \leq 10^9$ denotes the number of laps in the race,
- $0 \leq p \leq 10^9$ denotes how long it takes to change tires measured in seconds,
- $1 \leq r \leq 100$ denotes how many seconds it takes to race around the track on fresh tires, and

- $0 \leq b \leq 4$ denotes the tear factor on the wheels.

Output

A single integer, the minimum number of seconds required to finish the race.

Sample Input 1

```
10 10 100 2
```

Sample Output 1

```
1044
```

CPU Time limit 1 second

Memory limit 1024 MB

Authors Brigt Arve Toppe Håvardstun, Torstein Strømme, Petter Daae, and Simen Hornnes

Source Bergen Open 2021

License



Problem G

Glitching screen

Petter is taking the local train in a new city for the first time. He is well prepared and have memorized the names of all the stops. However, some of the pixels on the monitor that displays the upcoming stop are broken. Can you help him decide what the next stop is?

Input

The first line of input consists of three integers $1 \leq h \leq 100$, $1 \leq w \leq 100$ and $1 \leq n \leq 100$, the height and width of the screen in pixels and the number of train stops, respectively. Then there are $h \cdot n$ lines, where line $i \cdot h$ to $i \cdot h + h - 1$ shows how train stop i is rendered on a fully functioning screen (for stops $i \in \{0, 1, 2, \dots, n - 1\}$). The input ends with h more lines that shows the current screen. Pixels on the screen that are turned on are denoted by x and pixels that are turned off are denoted by a period ($.$). If a screen is broken, it means that one or more of the pixels are always off.

Output

Output yes if you can tell him what the next stop is, and no otherwise.



Unsplash license, Zürich Stadelhofen by

Ivan Heinzer via Unsplash

Sample Input 1

```

3 9 2
X.X.X.XXX
XXX.X..X.
X.X.X..X.
XX..X.XXX
X.X.X..X.
XX..X..X.
....X....
.X..X....
....X....

```

Sample Output 1

```

yes

```

Sample Input 2

```

1 4 2
X.XX
XX.X
X..X

```

Sample Output 2

```

no

```

CPU Time limit 1 second

Memory limit 1024 MB

Author Petter Daae

Source Bergen Open 2021

License



Problem H

100 meter dash

As CS student we sit around all day looking at screens. This is bad for us in many different ways; back pain, need of glasses, and lack of social life. According to the government regular exercise can help with some of these problems. Although you strongly believe this might not be the case, you have decided to start running, and tracking your runs with Ztrawa GPS-tracking mobile app.

Ztrawa stores your position coordinates (x, y) at some random intervals. When you are done with your run you can look at many different cool stats like your elevation gain, your fastest 100m, and your average speed from point A to B .

Since all CS students are looking to Min-Max everything you are only interested in your best 100m over your run. Sadly your Ztrawa app has a bug and the Min-Max feature has stop working, so you need to work it out yourself.

As Ztrawa only collect some finite amount of data points, we will assume constant speed between two consecutive data points. For example, say you were running around a 400m track and recorded 6 data points going all around the track and back where you started. This is sample case 1, where we started running from the bottom left, and went around counter-clockwise.

You need to create your own program so you can find what your fastest 100m was measured in seconds.

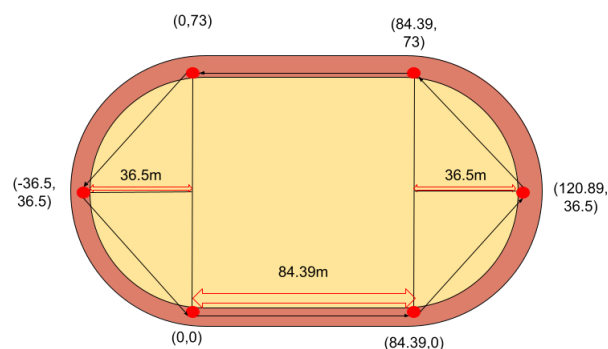


Illustration of sample case 1.

Input

The first line contains an integer $1 \leq n \leq 10^5$ indicating the number of readings from Ztrawa. Then follows n lines containing the information of each reading. The i^{th} such lines contains three real numbers x_i, y_i and t_i , indicating that at the time of reading i , you were at meter coordinates (x_i, y_i) relative to where you started, and t_i seconds has passed since the beginning of the session. It holds that $-10^5 \leq x_i, y_i \leq 10^5$ for every i , and $0 < t_1 < t_2 < \dots < t_n \leq 10^7$. All real numbers are given with at most 6 decimal places.

You always start running from $(0, 0)$ at time 0.

Output

Output a single real number, how fast you ran the fastest 100m measured in seconds. Anything with an absolute or relative error of 10^{-4} will be accepted.

Sample Input 1

```
6
84.39 0 10
120.89 36.5 14.3
84.39 73 18.4
0 73 28.5
-36.5 36.5 32.7
0 0 36.95
```

Sample Output 1

```
8.130299066033297
```

CPU Time limit

2 seconds

Memory limit

1024 MB

Author

Brigt Arve Toppe Håvardstun

Source

Bergen Open 2021

License



Problem I

Irritating accountants

You are shopping on behalf of a company for a big upcoming event and the pesky accountants ask you to sort the items you buy according to their own rigid standards. Why you ask? Nobody knows except for the accountants.



CC-BY 4.0,
FVM.145807
by
Flygvapenmuseum

Input

The first line of input contains two space-separated integers $1 \leq n \leq 10^5$ and $1 \leq k \leq 10^5$, indicating how many items you have bought and how many categories the accountants operate with respectively. On the second line follows n space-separated strings t_1, t_2, \dots, t_n , the names of the items you have bought. Some items may have been bought multiple times. On the third line follows c space-separated distinct strings c_1, c_2, \dots, c_k , the names of the categories in the order that the accountants require your items to be sorted.

Next follows k lines, one describing each category. The i^{th} such line begins with a string s_i , the name of the category it describes. It is followed by a positive integer m_i and then m_i space-separated distinct strings $t_1^i, t_2^i, \dots, t_{m_i}^i$, the items that belong to category s_i .

All strings in the input consists of between 1 and 10 characters from the English alphabet ([A-Za-z]). It is guaranteed that each item you bought belongs to exactly one category, and that c_1, c_2, \dots, c_k is a permutation of m_1, m_2, \dots, m_k . It also holds that $\sum_{i=1}^k m_i \leq 10^5$.

Output

On a single line, output n strings representing the items you bought in sorted order according to the accountants. If there are multiple ways of sorting the items according to the requirements, output any of them.

Sample Input 1

```
6 3
Bucket Milk Milk Cheese Rose
Drill
Groceries Flowers Hardware
Hardware 3 Bucket Drill Nail
Groceries 2 Milk Cheese
Flowers 2 Rose Tulip
```

Sample Output 1

```
Milk Milk Cheese Rose Drill
Bucket
```

CPU Time limit 3 seconds

Memory limit 1024 MB

Authors Petter Daae, Simen Hornnes, Torstein Strømme, and Kristoffer Æsøy

Source Bergen Open 2021

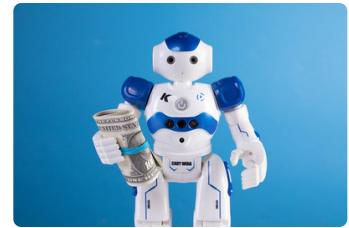
License



Problem J

Junior price robot

The idea behind your latest business adventure, BarGain Overview (BGO), is to collect the history of prices for a certain item that is available for sale on the web. The *BarGain score* of a particular day is defined as the number of days since the price was lower or equal to today's price. The worst BarGain score is thus 1, and if the price is strictly better than all previous prices ever recorded, then it is infinity. You want to report the BarGain score to your customers to help them identify a good buy.



CC-BY 2.0, Marco Verch via
Flickr

Input

The first line of input contains a single integer $2 \leq n \leq 400\,000$, the number of days for which you have collected price data for the item. On the second line of input follows n space-separated integers p_1, p_2, \dots, p_n , where $0 \leq p_i \leq 2 \cdot 10^6$ is the price of the item $i - 1$ days ago. Today's price is p_1 .

Output

A line containing today's BarGain score.

Sample Input 1

```
6
9 10 11 9 12 11
```

Sample Output 1

```
3
```

Sample Input 2

```
3  
1 2 3
```

Sample Output 2

```
infinity
```

CPU Time limit 1 second

Memory limit 1024 MB

Author Torstein Strømme

Source Bergen Open 2021

License



Problem K

King of cans

The informatics student Lisa at the University of Bergen, Norway, has spent way too much money. To make up for it, she has decided to recycle the empty cans she found in the reading hall. Since Lisa have been in Norway for a while now, she knows that the Norwegian government has a deposit-refund scheme to encourage recycling of soda bottles: when buying a bottle, the store will collect an extra 2 or 3 Norwegian kroner in deposit -- depending on bottle size -- for each soda bottle you buy. If someone returns an empty bottle to the store to be recycled, the deposit will be refunded to the redeemer. Hence, if Lisa manage to find a couple of hundred bottles left behind by wasteful students, it can quickly become surprisingly much money.



*Unsplash license, by Nick
Fewings on Unsplash*

As an informatics student, Lisa naturally has some tendencies of OCD and today it kicks in hard. She will not take any receipt not being exactly 100 Norwegian kroner after recycling the soda. The question is this: what is the maximum amount of receipts equaling exactly 100 Norwegian kroner Lisa can get?

Input

The first and only line have two space-separated integers x and y , where $0 \leq x \leq 10^4$ is the number of bottles whose deposit worth is 2 Norwegian kroner, and $0 \leq y \leq 10^4$ is the number of bottles whose deposit is worth 3 Norwegian kroner.

Output

Your program should output a single integer, the maximum number of refund receipts Lisa can get which each total exactly 100 Norwegian kroner.

Sample Input 1

5 30

Sample Output 1

1

Sample Input 2

49 1

Sample Output 2

0

Sample Input 3

0 100

Sample Output 3

0

CPU Time limit 1 second

Memory limit 1024 MB

Author Brigt Arve Toppe Håvardstun

Source Bergen Open 2021

License



Problem L

Live aid

You are organizing the annual “Boost the Global Opulence” fundraiser concert and have received an extraordinarily amount of requests from musicians that want to perform. However, you only have one stage, so you can not invite all of them.



*Unsplash License, Photo by
Anthony Delanoix via
Unsplash*

Each musician has a certain potential for attention, and you want to maximize the total attention potential for the concert.

But the musicians are very stubborn. They require that they can choose when their performance starts and for how long it lasts. More specifically, all of them have given you a start time and end time for their performance. If you cannot guarantee this, they will not perform at all.

Input

The first line contains a single integer $1 \leq n \leq 150\,000$, the number of musicians. Then follows n lines, one for each musician. Each line contains three space-separated integers s , e and a , where $0 \leq s \leq 10^6$ is the start time, $s < e \leq 10^6$ is the end time, and $0 \leq a \leq 10^6$ is the attention potential of the musician.

Output

Output the maximum total attention for the concert. In other words, pick a subset of the musicians to perform on the stage that maximize the sum of their attention potential, subject to the requirements that every musician gets their preferred time slot and there are no overlapping performances.

Sample Input 1

```
4
0 2 3
5 10 6
1 7 10
8 10 2
```

Sample Output 1

```
12
```

Sample Input 2

```
2
0 1 1
1 2 1
```

Sample Output 2

```
2
```

CPU Time limit 3 seconds

Memory limit 1024 MB

Author Petter Daae

Source Bergen Open 2021

License



Problem M

Meticulous smoothing

The arts and crafts teacher is looking at the beautiful plank you crafted in the woodshop, and gaze at you with a stern look. “This is not smooth enough! Use more sandpaper!”



*CC-BY-SA 4.0, Tim van de
Staak via Wikimedia
Commons*

Your plank is n cm long, and the arts and crafts teacher has measured the width of your plank on k different locations to prove his point. He demands that the thickness should differ by no more than 1 micrometer between any two consecutive measured location. If the sandpaper will shave off 1 micrometer of wood each time you use it at a particular location, how many times do you need to use the sandpaper?

Input

The first line of input contains a single integer $1 \leq n \leq 10^6$, the length of your plank. On the second line of input follows n space-separated integers k_1, k_2, \dots, k_n , the thickness of your plank ($1 \leq k_i \leq 10^6$ for every i).

Output

Output a single integer, the minimum number of times you need to use the sandpaper (assuming that the sandpaper only touch one location at the same time).

Sample Input 1

```
5  
1 6 7 2 5
```

Sample Output 1

```
10
```

CPU Time limit 1 second

Memory limit 1024 MB

Author Torstein Strømme

Source Bergen Open 2021

License

