United Kingdom and Ireland Programming Contest 2022

Problems

Problems are not ordered by difficulty.
Do not open before the contest has started.

This page is intentionally left (almost) blank.

Problem A
Alpine Pass

You are planning an expedition across the Pennines. You need to plan your route carefully to keep your team's strength up and avoid any unnecessary effort travelling high into the atmosphere. As such, you've obtained a survey map with geographical data showing the height profile of nearby terrain.

The terrain is represented by closed contour polygons which give you your altitude at the exact time you cross their boundary. If you are not currently on the boundary of a contour line, you can assume your altitude is equal to the smallest containing contour line.

To reach from point A to point B, what's the highest altitude you need to visit?

Input

- One line containing your integer start coordinates xy ($-10^6 \leq x, y \leq 10^6$).
- One line containing your integer end coordinates uv ($-10^6 \leq u, v \leq 10^6$).
- One line containing the number of contours n ($1 \leq n \leq 10^4$).
- n further lines, each containing the number of vertices in a contour v_i ($3 \leq v_i \leq 10^4$) and its height h ($0 \leq h_i \leq 10^6$) followed by v_i pairs of integers $x_{ij} y_{ij}$ giving a contour polygon ($-10^6 \leq x, y \leq 10^6$).

Contours do not overlap or touch each other or the endpoints. However, the largest contour line contains both of the endpoints. The total number of vertices in all contours does not exceed 10^4.

Output

Output the maximum altitude you will reach, if taking a route that keeps this maximum altitude as small as possible.

Sample Input 1

```
115 150
150 150
3
4 50 110 200 110 10 200 10 200 200
4 25 120 190 120 20 190 20 190 190
4 75 130 180 120 20 180 30 180 180
```

Sample Output 1

```
75
```

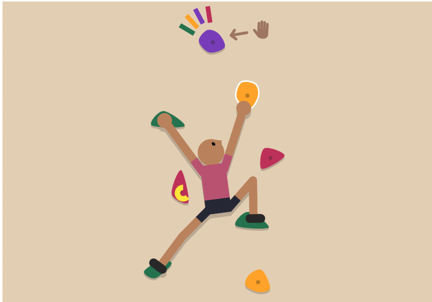
Sample Input 2

```
-100 20
50 -50
5
4 0 -200 -200 200 -200 200 200 -200 200
4 50 31 -44 31 -65 55 -62 60 -42
3 90 34 -47 34 -61 41 -52
6 20 -104 35 -72 18 -54 -48 65 -40 53 -80 -135 -58
3 30 -106 11 -129 -56 -56 -65
```

Sample Output 2

```
50
```

Problem B
Boulder Wall



You are opening a combined bouldering gym and bar with your business partner Cliff. As with any business venture, the key to success is having a strong product so you will be responsible for identifying the best routes.

Cliff already installed climbing holds of various colours on the flat vertical wall. In your gym, each route will consist of a sequence of holds no more than 100cm apart that go strictly up and then strictly down again ($y_1 < y_2 < ... < y_{m-1} < y_m > y_{m+1} > ...y_{n-1} > y_n$ for some m) while never touching the same hold more than once. Every route must start and end at or below the 100cm starting line.

Routes of the same colour can coexist on the same wall, however they must not overlap at all on the x axis. That is to say, if there are two routes of the same colour it must be possible to draw an infinite vertical line separating them.

Each hold on the wall has a quantifiable interestingness. Identify a set of routes that makes the sum of the interestingness of all holds used on routes as high as possible.

Input

- One line containing the number of holds, n ($1 \leq n \leq 100$), and the minimum height of the top of a route, h ($100 < h \leq 10000$).
- n further lines each containing the integer x and y coordinates of a hold ($0 \leq x, y \leq 10000$), its colour c as a lowercase Latin string ($1 \leq |c| \leq 10$), and its integer interestingness i ($1 \leq i \leq 1000$).

No two holds share the same (x, y) coordinates.

Output

Output the maximum interesting-ness you can create for the wall.

Sample Input 1

```
16 500
0 50 red 5
45 135 red 5
0 215 red 5
45 295 red 5
100 375 red 5
189 430 red 5
170 410 red 5
170 520 red 5
170 620 red 5
170 720 red 5
240 550 red 5
240 450 red 5
240 350 red 5
300 270 red 5
370 200 red 5
370 100 red 5
```

Sample Output 1

```
75
```

Sample Input 2

```
13 110
80 40 green 10
330 40 green 2
150 110 red 20
100 120 green 30
370 50 green 2
315 60 red 10
330 60 red 10
120 50 green 5
100 60 red 14
115 60 red 10
195 160 red 30
260 110 red 20
350 120 green 1
```

Sample Output 2

```
149
```

Sample Input 3	Sample Output 3
12 101 10 50 orange 1 12 48 orange 1 14 46 orange 1 16 44 orange 1 20 110 orange 1 22 110 orange 1 24 110 orange 1 26 110 orange 1 28 44 orange 1 30 46 orange 1 32 48 orange 1 34 50 orange 1	10

This page is intentionally left (almost) blank.

Problem C
Colour Wheel

Colours on computer screens are made up of red, green, and blue lights shining at different brightnesses. The ratio of the brightness of each light to the others determines the shade, and the total intensity determines the brightness.

For example, 3 lights shining at brightnesses 10:6:4 certainly don't shine as brilliantly as 3 lights shining at brightnesses 50:30:20, nonetheless they represent exactly the same shade.

Starting from complete blackness, we will add one unit of brightness to one of the red, green, or blue lights at a time. How many unique shades will we create this way?

Input

- One line containing the number of colour additions, n ($1 \leq n \leq 10^6$).
- One line of n words where word i contains the colour added at step i: one of `red`, `green`, or `blue`.

Output

Output the number of unique shades generated in this process, not counting the initial black colour.

Sample Input 1

```
3
red green red
```

Sample Output 1

```
3
```

Sample Input 2

```
6
red green blue blue green red
```

Sample Output 2

```
5
```

Problem D
Duck Crossing

Hook-a-duck is the traditional sport of floating rubber ducks in a tub, distributing long hooked poles to participants, and awarding points to participants for retrieving their duck.

Today we are organising the first world championship in hook-a-duck-ery. To facilitate the event, we used a local river instead of a single tub and floated each contestant's duck down to some distinct spot on the opposite bank. Now, the contestants will line up at their unique assigned places and try to latch their own duck fastest.

However, not everyone can retrieve their duck at the same time. If two participants would otherwise need to cross poles to fetch ducks, we will have to organise them into separate rounds.

How many rounds, at minimum, do we need to organise?

Input

- One line containing the number of participants, n ($3 \leq n \leq 10^5$).
- One line containing n distinct integers, the positions $p_{1\ldots n}$ of each participant along the near bank in metres ($1 \leq p \leq 10^6$).
- One line containing n distinct integers, the positions $d_{1\ldots n}$ of each duck along the other bank in metres ($1 \leq d \leq 10^6$).

Output

Output the minimum number of rounds we need to organise to accommodate all of the contestants and their ducks.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

```
5
42 22 54 19 37
14 15 26 13 10
```

Sample Output 2

```
3
```

Problem E
Eager Packing

As CTO (Chief Transport Officer) at your new logistics company, one of your duties is to bring the packages of various sizes to the loading area to supply the delivery vehicles for the day.

The vehicles load up one-by-one. Each vehicle is loaded automatically by a machine that finds the largest package that can fit in the vehicle without over-weighting, and loading it on. This repeats until no more packages can fit and the next vehicle approaches.

Knowing the number of vehicles, and the sizes of packages you have available, what is the smallest weight limit per vehicle that you can set while ensuring you distribute all of the packages?

Input

- One line containing the number of packages, n $(1 \leq n \leq 10^5)$, and the number of vehicles, v $(1 \leq 10^5)$.
- One line containing n integers: the size of each package $s_{1...n}$ $(1 \leq s \leq 10^5)$.

Output

Output the smallest weight limit per vehicle that you can set in order to still distribute all of the packages to vehicles without over-weighting.

Sample Input 1	Sample Output 1
6 4 1 1 1 1 1 1	2

Sample Input 2	Sample Output 2
9 3 31 15 2 3 4 17 22 38 12	49

Problem F
Fungible Growth

After the beating your technology investments took in 2022, you're going to try to recoup some of your losses by investing in safer strategies. Namely, you've been given tips on two different investment strategies and you want to pick the one that will give the highest return.

Out of the two investment strategies, knowing their expected increases after a year, which one should you invest in for higher returns?

Input

- One line containing the expected return on investment A, a ($1.0 \le a \le 10^6$).
- One line containing the expected return on investment B, b ($1.0 \le b \le 10^6$).

The values of a and b are guaranteed to differ by at least 10^{-6}.

Output

Output `A` if you should invest in strategy A, or `B` if you should invest in strategy B.

Sample Input 1

```
200.00
150.11
```

Sample Output 1

```
A
```

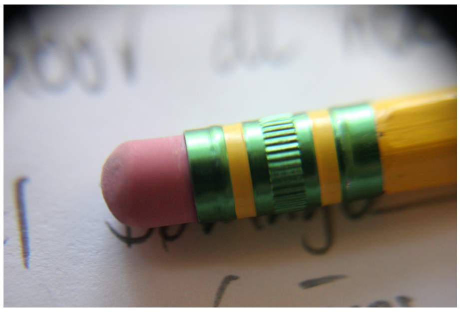
Sample Input 2

```
9999.99
10000
```

Sample Output 2

```
B
```

Problem G
Grapheme Game



A word is written on a board and then you and your opponent take turns to shorten the word by removing one or more letters from the start and/or end to form new words. Each letter of the alphabet is assigned a point score, and if you delete a letter you get its points added to your score.

Your goal is to receive as small a score as possible. What is the lowest score you can get if your opponent plays well?

Input

- One line containing the lowercase string S ($1 \leq |S| \leq 10^5$).
- One line containing the 26 integer scores s_i for each letter of the alphabet in order ($1 \leq s \leq 10^4$).
- One line containing the size of the dictionary, n ($1 \leq n \leq 10^5$).
- n further lines each containing a unique dictionary word w_i ($1 \leq \sum(|w|) \leq 10^6$).

Output

Output the smallest point score you can get by playing the game optimally, assuming your opponent plays optimally too.

Sample Input 1

```
unredeemable
1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6
6
redeem
redeemable
able
deem
a
red
```

Sample Output 1

```
22
```

Sample Input 2

```
hamsandwich
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2
ham
sand
```

Sample Output 2

```
7
```

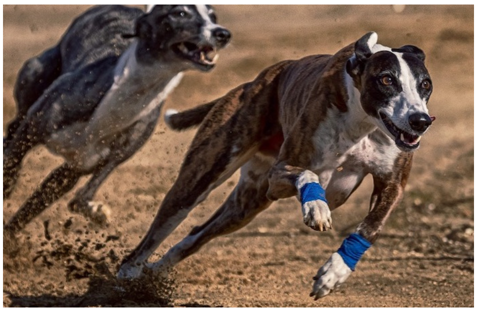
Sample Input 3

```
banana
1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5 20
4
a
an
nana
ban
```

Sample Output 3

```
5
```

Problem H
Hare's Breadth



You have taken on a new job as photographer at a greyhound track. Your job is to take the best pictures of the crowd favourites, the winners, and even the underdog last-place finshers in the Greyhound Prix.

In the Greyhound Prix, runners are lined up behind the start line at various positions, the order of this starting line-up being determined by previous races.

Today you'll be taking a photo of the entire race pack. To get a good picture you must have the entire pack as close as possible, with minimal distance between first place and last at the time you take the picture. If you time it right (including before the race has started), how close together can you hope for?

Input

- One line containing the number of runners, n ($1 \le n \le 10^6$).
- One line containing n real numbers, the start places of each runner $s_{1 \ldots n}$, in metres ($-10^6 \le s \le 0$).
- One line containing n real numbers, the speed of each runner $v_{1 \ldots n}$, in metres per second ($-10^6 \le v \le 0$).

Output

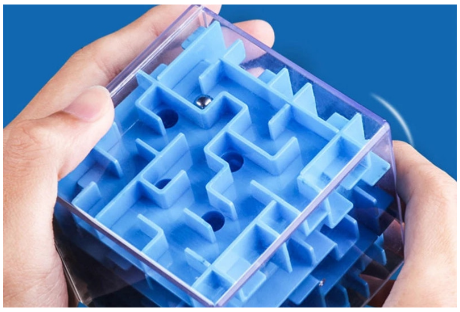
Output the closest distance you can achieve between first and last place at any time to get a good photo.

Sample Input 1	Sample Output 1
2 0 -10 10 5	0

Sample Input 2	Sample Output 2
3 -20 -100 -10 50 51 50	10

Sample Input 3	Sample Output 3
3 0 -1 -2 1.0 3.0 4.0	0.3333333333333

Problem I
Incline Maze



A popular pocket game involves guiding a ball through a maze by turning and letting gravity do the work. Whenever the board is tilted in one of the cardinal directions, the ball moves quickly as far as it can in that direction and then comes to rest, unless it passes over the goal hole in the way, in which case it falls in and the game is over.

What is the minimum number of times you need to tilt the board and change the direction of the ball in order to win such a game?

Input

- One line containing the number of rows and columns r c of the game ($1 \leq r, c \leq 10^3$).
- r further lines each containing c characters:
  - `.` for an empty space into which the ball can roll.
  - `x` for a blocked cell against which the ball comes to rest.
  - `s` for the starting location of the ball.
  - `t` for the desired end location of the ball.

The board always contains exactly one `s` and `t` and always has a border consisting of `x` blocks.

Output

Output the minimum number of tilt moves necessary to complete the game, or `impossible` if the goal cannot be reached with any number of moves.

Sample Input 1

```
6 8
xxxxxxxx
x.x....x
x....x.x
x....x.x
x..xtxsx
xxxxxxxx
```

Sample Output 1

```
5
```

Sample Input 2

```
6 8
xxxxxxxx
x......x
x....x.x
x....x.x
x..xtxsx
xxxxxxxx
```

Sample Output 2

```
impossible
```

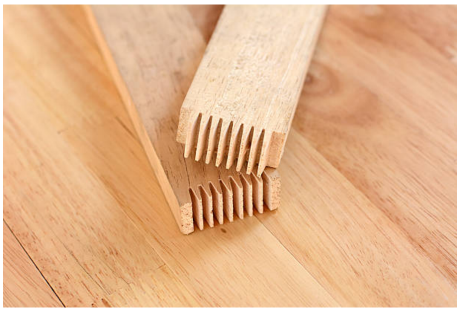
Sample Input 3

```
5 5
xxxxx
x..tx
x.s.x
x...x
xxxxx
```

Sample Output 3

```
2
```

Problem J
Joinery



For an upcoming woodworking project you need a plank of a certain length. As you don't have one available, you're willing to fasten together two shorter wooden planks to create something of the right length or greater, and then cut the excess down to size.

Assuming you can find two such planks, what is the minimum amount of excess plank you will need to cut off?

Input

- One line containing the length of plank you need to make, l ($2 \leq l \leq 10^9$).
- One line containing the number of smaller planks you have, n ($2 \leq n \leq 10^6$).
- One line containing n integers, the lengths of smaller planks $p_{1 \ldots n}$ ($1 \leq p < l$).

Output

If you can find two planks to join and then cut down to size, output the number giving the smallest extra length you can generate by doing so.

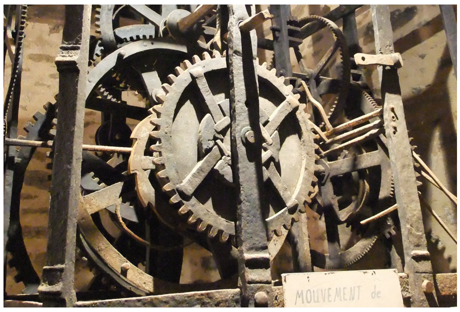
Otherwise, output `impossible`.

Sample Input 1	Sample Output 1
10 4 6 8 7 9	3

Sample Input 2	Sample Output 2
100 4 25 30 30 25	impossible

Sample Input 3	Sample Output 3
10 2 6 4	0

Problem K
Keeping Time



Although most clocks move at a perceptible speed, others can be very fast or very slow. For example, it is possible to build a clock that ticks once every millenium by linking together multiple axles with very large gear ratios.

For example, an initial axle might hold a gear with 10 teeth. This may drive another gear with 400 teeth on a second axle, which moves 40 times slower in the opposite direction. Then that axle may have another 20-toothed gear meshing with a 400-toothed gear on a third axle, giving a 20-fold slowdown between the two and a 800-fold slowdown overall. With more gears the ratio between fastest and slowest gear can go into the millions and billions.

We noticed that one such millenium clock has an unusual mechanism with redundant gears indirectly connecting the same axles in multiple ways. For the system to actually work, all of these connections must have the same effective gear ratio and lead to the gears really turning. Is this the case?

Input

- One line containing the number of gears, n ($1 \leq n \leq 10^6$).
- One line containing the number of teeth on each gear, $g_{1\ldots n}$ ($1 \leq g \leq 10^6$).
- One line containing the number of shared axles, s ($1 \leq s \leq n$).
- m lines, each containing the indices of two gears a, b sharing an axle ($1 \leq a, b \leq n$).
- One line containing the number of meshing gear pairs, m ($1 \leq m \leq n$).
- m lines, each containing the indices of two gears a, b that mesh ($1 \leq a, b \leq n$).

If no shared axle is specified for a gear, it is assumed that this gear has its own axle.

Output

If the gears can all turn, output `yes`. Otherwise, output `no`.

Sample Input 1

```
6
10 20 30 60 40 20
2
3 4
5 6
4
1 2
2 3
4 5
6 1
```

Sample Output 1

```
yes
```

Sample Input 2

```
3
10 10 10
0
3
3 2
2 1
1 3
```

Sample Output 2

```
no
```

Problem L
Lasso Tool

You are designing the "magic lasso" tool for a new farming simulator app. This tool, given the locations of one or more fields full of sheep, draws a set of closed fences around those fields so that the sheep cannot escape out of the game world boundaries.

In this isometric world, fences can be aligned at 0, 45, or 90 degrees from the horizontal. That is to say, they can travel along the edge of a field or diagonally through it.

What is the smallest total length of fence you need to prevent any of the sheep from escaping the game board?

Input

- One line containing the number of rows and columns r and c ($1 \leq r, c \leq 100$).
- r lines, each with c characters representing empty space (`.`) or sheep-filled space (`x`).

Output

Output the minimum line length required to enclose all of the sheep-filled cells inside loops made from horizontal, vertical, and diagonal lines, possibly encompassing more than one region.

The output must be accurate to an absolute or relative error of at most 10^{-6}.

Sample Input 1

```
4 8
......xx
.......x
........
..xx....
```

Sample Output 1

```
13.41421356237
```

Sample Input 2

```
7 7
.......
.xx.xx.
.x...x.
...x.x.
.x...x.
.xxxxx.
.......
```

Sample Output 2

```
20
```

Sample Input 3

```
3 3
x..
x..
.xx
```

Sample Output 3

```
10.24264068712
```