Monash Collegiate Programming Contest

25th, April, 2020

FQAs

Q: Why I passed all samples, but still get `Wrong Answer (WA)` ?

- There are multiple secret test cases, failing on any of them causes `Wrong Answer`, so you'll have to spot the bug by yourself.

Q: What is `Time Limit Exceeded (TLE)` ?

- Your program runs out of time limit; this error doesn't allow you to know if your program would reach the correct solution to the problem or not.

Q: What is `Memory Limit Exceeded (MLE)` ?

- Your program tried to use more memory than the judge allows.

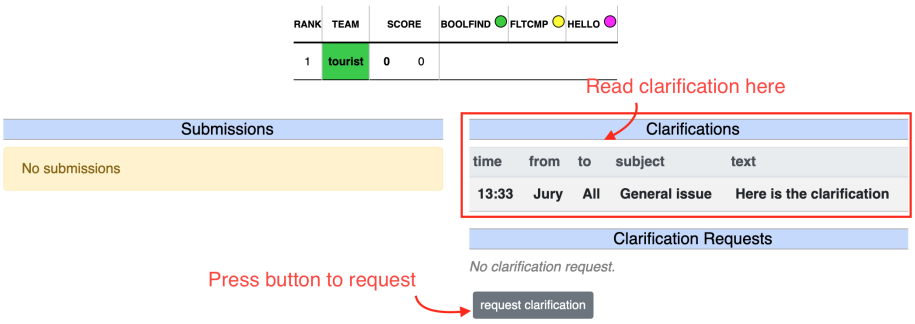
Q: What is `Runtime Error (RE)`

- Your program failed during the execution (segmentation fault, floating point exception...). The exact cause is not reported to the user to avoid hacking.

Q: There are `Input file: standard input`, `Output file: standard output` in my problem statement, what do these mean? Do I need to create files?

- Reading from standard input (`stdin`) means reading data which you are typing on your keyboard, and outputting to standard output (`stdout`) means printing data to your console. So it's not necessary to create files.

Q: How to read and request clarifications?



-

Tips

- Pay attention to `Time limit`, `Memory limit` constraints in your problem statement when you get `Time Limit Exceeded` or `Memory Limit Exceeded` verdict.
- `C++` can nearly run 10^8 basic operations (e.g. `+,-`) in 1 second, `Python` can nearly run 10^6 basic operations in 1 second, and you can esitmate your program running time based on these info and your compexity analysis to avoid `Time Limit Exceeded`.
- Instead of typing on keyboard, a more convenient way for local testing on command line is:

```
python myprogram.py < sample.in > myoutput
diff myoutput sample.out
```

  where `<` is redirect all contents of file `sample.in` to `standard input`, and `>` is redirect your `standard output` to the file `myoutput`, and `diff` is to check the difference between your output and the expected answer.

- If you are going to run your script `myscript.sh` on command line, make sure it has execute permission

```
chmod u+x myscript.sh
./myscript.sh
```

  otherwise, you may get error message `permission denied: ./myscript.sh`.

- If your want to run your python script `myscript.py` by `./myscript.py`, make sure you've added following line at the top of your python file

```
#!/usr/bin/env python

def func():
  ...
```

Good Luck & Have Fun!

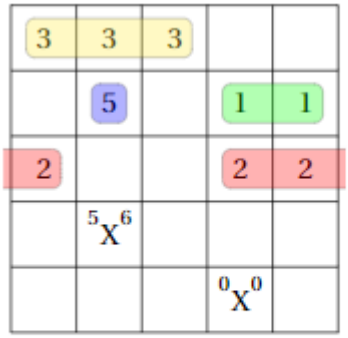
Problem A. Hearty Frogger I

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

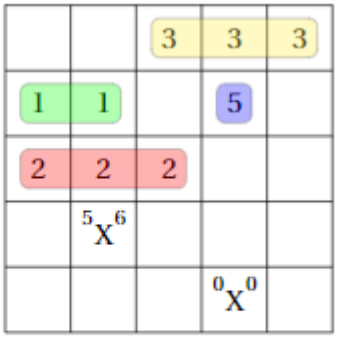
Note that the statement is same as "Hearty Forgger II", except for constraints

Alice is playing a variant of *Frogger*.

This game is played on a $b \times b$ grid, indexed by (0, 0) from the bottom left, with the x axis spanning left to right. On the grid, there are a number of horizontal, possibly overlapping, single height trucks, of varying lengths, which contains *hearts*.



Case 0 at Time 0.



Case 0 at Time 2.

Every time-step, the trucks move 1 unit to the right, wrapping around the board from right to left. Simultaneously, the player will move one unit up the grid, until falling off the top of the grid, at which point the game is over.

Whenever the player is on top of a truck (including when the player spawns, at its starting row), the player earns the amount of hearts contained within the truck.

Therefore, the only input Alice has into the outcome of the game is the player's spawn, as well as the time at which this spawn occurs.

Alice has only a few options for such spawn locations and times, and she wants to maximize the amount of hearts gained before she finishes the game. Can you help her?

Input

The first line contains 3 integers: b, p, q.

p lines follow, denoting the locations of trucks on the map. Each line contains 4 integers: x_i, y_i, l_i, h_i.

x_i, y_i denotes the location of the left side of the truck, while l_i denotes the length of the truck. h_i denotes the total heart value of this truck.

After this, q lines follow. Each line contains 4 integers: $x_j, y_j, t1_j, t2_j$.

This denotes a singular spawn point, as well as a start and end time ($t1_j$ and $t2_j$ respectively, both inclusive) under which Alice can spawn here.

Constraints

$1 \leq b \leq 50$.

$0 \leq p, q \leq 3b$

$0 \leq x_i, y_i, x_j, y_j < b.$

$0 \leq t_1 \leq t_2 \leq 10^8$

$0 < l_i \leq b$ $0 < h_i \leq 10^4$

Output

h, the maximum number of hearts achievable from one spawn location at one time, given the above information.

Example

standard input	standard output
5 4 2 3 2 3 2 3 3 2 1 1 3 1 5 0 4 3 3 3 0 0 0 1 1 5 6	6

Note

Case 0 can achieve 6 hearts by beginning at (1, 1) at time 6.

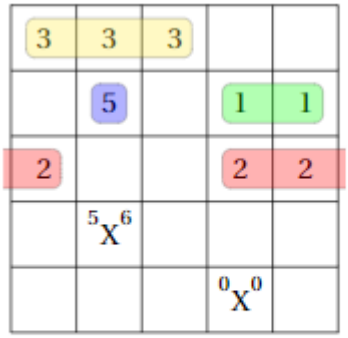
Problem B. Hearty Frogger II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 megabytes

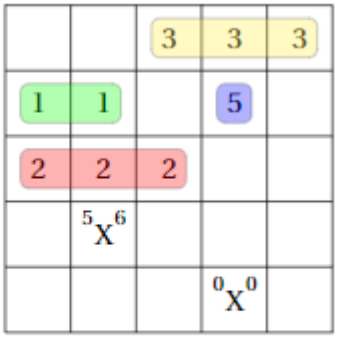
Note that the statement is same as "Hearty Forgger I", except for constraints

Alice is playing a variant of *Frogger*.

This game is played on a $b \times b$ grid, indexed by (0, 0) from the bottom left, with the x axis spanning left to right. On the grid, there are a number of horizontal, possibly overlapping, single height trucks, of varying lengths, which contains *hearts*.



Case 0 at Time 0.



Case 0 at Time 2.

Every time-step, the trucks move 1 unit to the right, wrapping around the board from right to left. Simultaneously, the player will move one unit up the grid, until falling off the top of the grid, at which point the game is over.

Whenever the player is on top of a truck (including when the player spawns, at its starting row), the player earns the amount of hearts contained within the truck.

Therefore, the only input Alice has into the outcome of the game is the player's spawn, as well as the time at which this spawn occurs.

Alice has only a few options for such spawn locations and times, and she wants to maximize the amount of hearts gained before she finishes the game. Can you help her?

Input

The first line contains 3 integers: b, p, q.

p lines follow, denoting the locations of trucks on the map. Each line contains 4 integers: x_i, y_i, l_i, h_i.

x_i, y_i denotes the location of the left side of the truck, while l_i denotes the length of the truck. h_i denotes the total heart value of this truck.

After this, q lines follow. Each line contains 4 integers: $x_j, y_j, t1_j, t2_j$.

This denotes a singular spawn point, as well as a start and end time ($t1_j$ and $t2_j$ respectively, both inclusive) under which Alice can spawn here.

Constraints

$1 \leq b \leq 10^5$.

$0 \leq p, q \leq 3b$

$0 \leq x_i, y_i, x_j, y_j < b.$

$0 \leq t_1 \leq t_2 \leq 10^8$

$0 < l_i \leq b$ $0 < h_i \leq 10^4$

Output

h, the maximum number of hearts achievable from one spawn location at one time, given the above information.

Example

standard input	standard output
5 4 2 3 2 3 2 3 3 2 1 1 3 1 5 0 4 3 3 3 0 0 0 1 1 5 6	6

Note

Case 0 can achieve 6 hearts by beginning at (1, 1) at time 6.

Problem C. Fast and Furious I

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

*Note that the statement is same as "Fast and Furious II", except for **output** and **constraints***

The road network is an **infinite** 4-connected grid map, Farmer John is driving his broken car from $(s_x,\ s_y)$ to repair station $(t_x,\ t_y)$. The start location and repair station are in different places, and $t_x \geq s_x, t_y \geq s_y$. At the time 0, Farmer John is at $(s_x,\ s_y)$ and start driving, the speed of the car is 1 cell per second. Due to some mechanical issues, the broken car **can not stop** until reaching the repair station, and Farmer John has to change the direction at every m seconds. In other words, the direction of his path should be different at seconds $k * m$, $k * m + 1$ where $k > 0$. For example, he can drive in whatever direction from 1 to m seconds, the driving direction at the $m + 1$ second must be different from the m second, and so on.

What's the minimum time for Farmer John to reach the repair location?

Input

The first line contains two integers m $(1 \leq m \leq 100)$.

The second line contains two integers $s_x\ s_y$ $(1 \leq s_x, s_y \leq 100)$, the coordinate of the start location.

The third line contains two integers $t_x\ t_y$ $(1 \leq t_x, t_y \leq 100)$, the coordinate of the repair station.

Output

The first line contains an integer T, the minimum time to reach the repair location. In the next T lines, the i-th line contains two integers $x\ y$, the coordinate of the i-th seconds. If there are multiple valid shortest paths, you can output any of them.

Examples

standard input	standard output
2 1 1 1 4	5 1 2 1 3 0 3 0 4 1 4
1 1 1 1 4	5 1 2 0 2 0 3 1 3 1 4

Problem D. Fast and Furious II

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

*Note that the statement is same as "Fast and Furious I", except for **output** and **constraints**.*

The road network is an **infinite** 4-connected grid map, Farmer John is driving his broken car from $(s_x,\ s_y)$ to repair station $(t_x,\ t_y)$. The start location and repair station are in different places, and $t_x \geq s_x, t_y \geq s_y$. At the time 0, Farmer John is at $(s_x,\ s_y)$ and start driving, the speed of the car is 1 cell per second. Due to some mechanical issues, the broken car **can not stop** until reaching the repair station, and Farmer John has to change the direction at every m seconds. In other words, the direction of his path should be different at seconds $k * m$, $k * m + 1$ where $k > 0$. For example, he can drive in whatever direction from 1 to m seconds, the driving direction at the $m + 1$ second must be different from the m second, and so on.

Farmer John is wondering how many different paths to the repair station with the minimum time?

Output the minimum time and the number of possible ways modulo 1000000007.

Input

The first line contains two integers m $(1 \leq m \leq 50)$.

The second line contains two integers s_x s_y $(1 \leq s_x, s_y \leq 50)$, the coordinate of the start location.

The third line contains two integers t_x t_y $(1 \leq t_x, t_y \leq 50)$, the coordinate of the repair station.

Output

The output contains only one line, two integers t cnt, the minimum time and number of possible paths (modulo 1000000007) with the corresponding time.

Examples

standard input	standard output
1 1 1 1 4	5 2
2 1 1 1 4	5 8

Note

In the first sample, the minimum time cost is 5, there are two paths:

- (1, 1) → (1, 2) → (0, 2) → (0, 3) → (1, 3) → (1, 4)
- (1, 1) → (1, 2) → (2, 2) → (2, 3) → (1, 3) → (1, 4)

Problem E. Storage Room I

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

Note that the statement is same as "Storage Room II", except for constraints.

A new storage room is going to be built for the newly produced high-tech devices that are going to be released soon.

However, since we have spent so much money on research and development of this high-tech device, we need to build this room with the minimum cost.

We already know the exact number of devices we need to store and thus, we know the exact volume for this room: V. Therefore, the only cost left to minimize is painting the room (we can't leave the room not painted obviously).

A room is a cuboid with a length, width and height. It has 6 sides that are rectangles and any pair of sides facing each other have exactly the same shape and size. The volume of the room is calculated as $L * W * H$ where L is the length, H is the height and W is the width.

In order to paint this room, we need to paint all of the 6 sides (4 walls, the floor and the ceiling). Painting each square meter of the room costs us 1 dollar.

This room can have any integer length, width and size as long as the volume is equal to V. What is the minimum cost for painting such a room?

Input

The first line of input contains an integer V: the volume this new room should have.

$1 \leq V \leq 10^3$.

Output

In the only line of output, print an integer: the minimum cost for painting a room with volume V.

Examples

standard input	standard output
2	10
5	22
10	34

Note

You may set the length, width and height of the room (cuboid) to be whatever integer you want. The only constraint is that the volume of the room should be V.

Problem F. Storage Room II

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

Note that the statement is same as "Storage Room I", except for constraints.

A new storage room is going to be built for the newly produced high-tech devices that are going to be released soon.

However, since we have spent so much money on research and development of this high-tech device, we need to build this room with the minimum cost.

We already know the exact number of devices we need to store and thus, we know the exact volume for this room: V. Therefore, the only cost left to minimize is painting the room (we can't leave the room not painted obviously).

A room is a cuboid with a length, width and height. It has 6 sides that are rectangles and any pair of sides facing each other have exactly the same shape and size. The volume of the room is calculated as $L * W * H$ where L is the length, H is the height and W is the width.

In order to paint this room, we need to paint all of the 6 sides (4 walls, the floor and the ceiling). Painting each square meter of the room costs us 1 dollar.

This room can have any integer length, width and size as long as the volume is equal to V. What is the minimum cost for painting such a room?

Input

The first line of input contains an integer V: the volume this new room should have.

$1 \leq V \leq 10^{12}$.

Output

In the only line of output, print an integer: the minimum cost for painting a room with volume V.

Examples

standard input	standard output
2	10
5	22
10	34

Note

You may set the length, width and height of the room (cuboid) to be whatever integer you want. The only constraint is that the volume of the room should be V.