All-Ireland Programming Olympiad 2021

March 20, 2021

Contributors

The AIPO organisers would like to thank the following people.

For leading the question writing:

- Dr Sabin Tabrica
- Marcu Bogdan Stefan
- Colm Hickey
- Andrew Nash

For reviewing and editing the questions:

- Brian McCarthy
- Jakub Piatek

1 Some Curious Primes

Primus has just learned to convert a number between bases through consecutive divisions. As Primus is very fond of prime numbers, they started to decompose the regular base-10 primes in all the other bases from 2 to 10. For some primes, Primus observed that the sum of their digits may also be prime across all bases from 2 to 10 and they called them "curious primes".

Given a number N then check if it is curious prime or not.
Output "YES", if N is a curious prime, and "NOT" otherwise.

Input/Output

A single integer, N

Examples

Sample Input 1

```
13
```

Sample Output 1

```
NOT
```

Sample Input 2

```
131
```

Sample Output 2

```
YES
```

Explanation of Sample Input 1

$$(13)_{10} = (1101)_2 = (111)_3 = (31)_4 = (23)_5 = (21)_6 = (16)_7 = (15)_8 = (14)_9$$

And, $1 + 3 = 4, 3 + 1 = 4$ and $1 + 5 = 6$ (for bases 10,4,8) are all non-prime, and so violate the conditions for curious primes.

Constraints

- $0 \leq N, C \leq 2\,000\,000\,000$.

2 Busy Marina

The marina in a small village is very badly organized with boats anchored randomly in either vertical or horizontal positions, relative to the land. Aerial photography has given a clear image of how the boats have been chaotically placed in the marina.

John, the marina manager, has reduced the photo to a matrix with N rows and M columns in which the boats are represented by consecutive sequences of 1's and the water by ares of 0's. A boat is represented by at least 2 consecutive 1's, depending on its size - smaller boats are represent fewer 1's and longer boats with many consecutive of 1's.

The boats are all placed such that they do not make contact, or collide, i.e. a vertical boat cannot touch a horizontal boat.

Note that there may be in the map some noise, created by John's image compression, that may result in some isolated 1's (surrounded by 0's on all four sides) which does not represent a boat. Given such an N by M matrix, determine how many boats are anchored in the marina

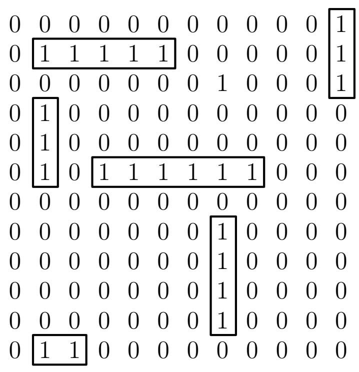


Figure 1: An example 12x12 marina, with the six boats indicated

Input/Output

The first two lines contain two integers, N and M, corresponding to the height and width of the matrix

The next N lines contain M space separated 0's and 1's.

Output the number of boats in the matrix

Examples

Sample Input 1

```
12 12
0 0 0 0 0 0 0 0 0 0 0 1
0 1 1 1 1 1 0 0 0 0 0 1
0 0 0 0 0 0 0 1 0 0 0 1
0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0 0
```

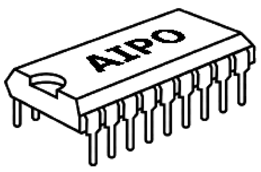
Sample Output 1

```
6
```

Constraints

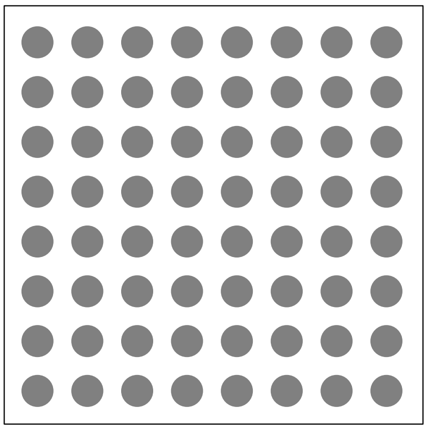
- $0 \leq N, M \leq 1000$

3 Largest LGA Chip Placement



You are a penetration testing specialist that has been tasked with exfiltrating data from a client's data centre. After cloning the badge of a security guard, you have succeeded in gaining access to the central server room.

Unfortunately there are no peripherals or ports that allow you to interact with the servers. Out of the corner of your eye, you spot a server that has its motherboard with a land grid array (LGA) socket exposed. You know that you can exploit this to achieve your goal.


(a) An example of a Land Grid Array socket

```
0V 5V 5V 5V 0V 0V 0V 5V
0V 0V 0V 5V 0V 0V 5V 0V
5V 5V 5V 5V 5V 0V 0V 0V
0V 0V 0V 5V 0V 5V 5V 0V
0V 0V 5V 0V 0V 5V 5V 0V
5V 5V 0V 5V 5V 5V 5V 0V
5V 0V 5V 0V 5V 5V 5V 0V
5V 0V 0V 5V 0V 5V 5V 0V
```

(b) An example power state of an 8x8 LGA socket

Using your portable volt-meter you have established which of the pads are powered and which ones are drains (have no power).

You have to find the area of the biggest chip that you can place on the board without causing a short circuit. A short circuit is caused by your chip if the amount of voltage pads and non-voltage pads that it covers is not equal. You have access to chips of any rectangular shape.

Input/Output

The first two lines contain two integers, N and M, corresponding to the height and width of the LGA socket.

The following N lines contain M integers, the voltage level V_{ij} at that position on the socket

Output the size of the maximum area containing an equal number of 5's and 0's. If there is nowhere to place the chip without causing a short circuit, you should output "No possible chip"

Examples

Sample Input 1

```
4 4
0 0 5 5
0 5 5 0
5 5 5 0
5 0 0 5
```

Sample Output 1

```
8
```

Sample Input 2

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

Sample Output 2

```
6
```

Sample Input 3

```
1 1
0
```

Sample Output 3

```
No possible chip
```

Constraints

- $0 \leq N, C \leq 200$.
- $0 \leq V_{ij} \leq 5$

4 Massive Watermain Misspending

As a project manager assigned to building a new town (in a very ordered society), you have paid a contractor to lay piping for the town's water network. The town has N main water exchanges. Water will enter one of these from the national pipeline, and return to the national pipeline from a different exchange. These are known as the source and drain. Alas! The contractor did a terrible job - they laid far more pipes than you needed, and never connected to the national pipeline.

We need to finish the job ourselves - select the pipes that we want to use for our network, and connect our source and drain to the national pipeline, while maximising the flow in out network.

To comply with regulations, every exchange must be connected in a specific order, as a loop. Note that if an exchange is designated as a drain, the previous exchange must be designated as a source, and the two exchanges do not have to be connected.

Pipes

- Pipes between exchanges are connected by junctions which have unique integer numbers greater than N
- There are no other guarantees about the values, or order of the numbers of the junctions, other than that they are all unique.
- It is guaranteed that a pipe will lie on a direct path (one that does not have to pass through another exchange), between exactly two exchanges.

Flow

- The flow between two exchanges is the minimum capacity of the pipes that the flow is through.
- The flow of the network is the minimum of all the flows between any exchanges.

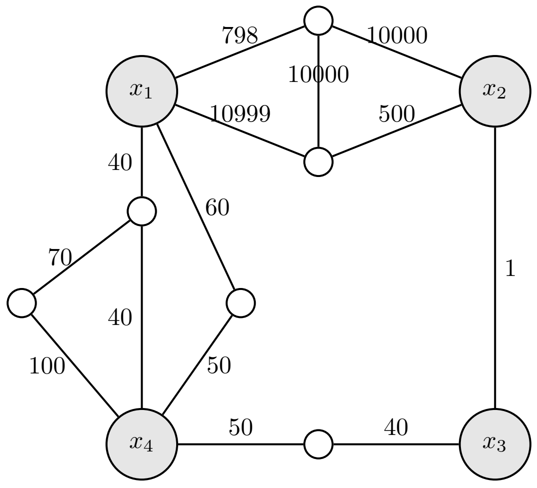


Figure 3: An example network of pipes. Grey nodes denote the 4 main exchanges

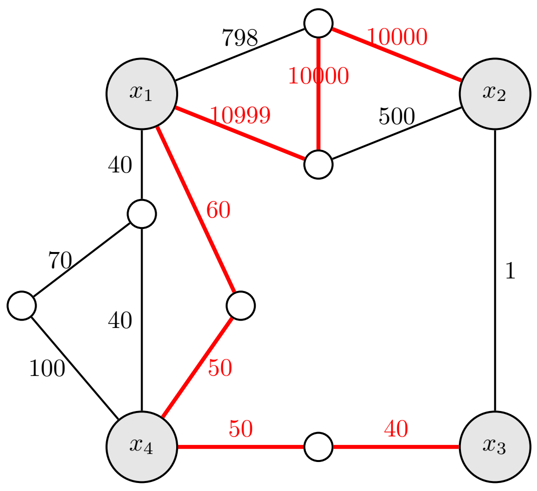


Figure 4: The optimal solution for the above network. Nodes x_2 and x_3 are designated source and drain respectively. The overall capacity of this network is 40, which is minimal.

Input/Output

The first line of input is N, the number of exchanges.

Following this, we repeat the following procedure from $i = 1, \ldots N$, to read the network of pipes between exchanges i and $i + 1$

Read a integers j, p denoting the number of junctions (points where two or more pipes meet) and pipes between exchanges i and $i + 1$.

There then follow p lines, each containing 3 integers, a, b and w - this represents a pipe with capacity w running from junction/exchange a to junction/exchange b.

Output 3 integers, the source, the drain, and the capacity of the completed network.

Sample Input 1

```
4
2 5
1 5 798
5 6 10000
1 6 10000
5 2 10000
6 2 500
0 1
2 3 1
1 2
3 7 40
7 4 50
3 6
4 8 100
4 10 40
4 9 50
9 1 60
8 10 70
10 1 40
```

Sample Output 1

```
2 3 40
```

This example corresponds to the input presented above.

Constraint

- $1 \leq N \leq 4\,000$
- $1 \leq \text{junctions} \leq 500$
- $1 \leq \text{pipes} \leq 125\,000$

5 Planned Production Line Problems

The manager of an electric vehicle manufacturing plant has become too preoccupied with ambitions of space colonisation, to have time to monitor their cutting edge AI robotic car assembly line.

You have been tasked with implementing a system to track and report on the status of the different robotic arms on the line.

The arms are arranged sequentially along a conveyer belt, and are numbered from $0, 1, \dots n$. Each arm has an Operational Percentage Indicator (OPI), a value from 0 to 100 that indicates to what extent it is working at full capacity.

Your system has to be able to handle the following queries, given an initial sequence of OPIs.

When an arm fails, we no longer consider its OPI when examining the OPIs of the arms along the line. Further, after a failure, the indices of all arms down-line from the failure are decreased by one.

An engineer has speculated that failures are more likely between adjacent machines, possibly due to climactic conditions.

- Set the OPI of arm x to value O
- Get the mean, and minimum OPI of all arms in range (a, b)
- Handle the failure of arm x
- An engineer has speculated that failures are more likely between adjacent machines, possibly due to climactic conditions. We occasionally want to check if the (operational) arm at index i is adjacent to a failed arm

Example

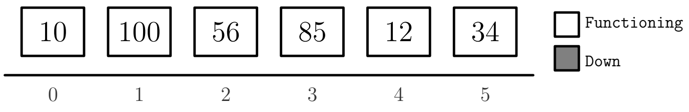


Figure 5: Initial configuration of the production line, showing 6 arms with different OPIs.

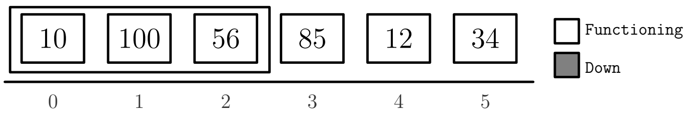


Figure 6: Querying range (0,2).

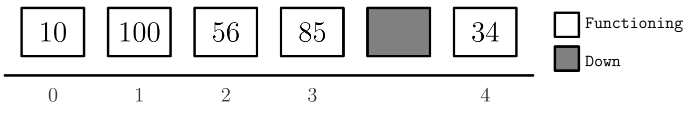


Figure 7: Arm 4 reports failure.

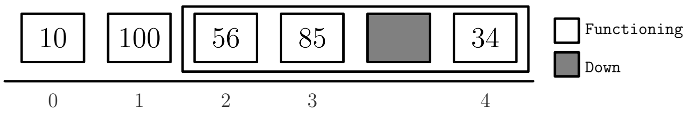


Figure 8: Querying range (2,4).

Input/Output

The first line of input is a single integer N, the length of the assembly line. This is followed by N integers between 0 and 100, the OPIs of each arm on the line.

This is followed by a single integer Q, and then Q lines where

- If the first character of the line is F, it will be followed by a single integer - the index of the arm that has reported failure.
- If it is a Q, then you need to perform a query between the two indices that follow. You should print the result of the query to the standard output, as two space separated integers (average followed by minimum of the range).
- If it is a C, we need to check if the arm at the given index is adjacent to a failed arm. If so, print the string YES to the standard output, otherwise print NO

Examples

Sample Input 1

```
6
10 100 56 85 12 34
5
Q 0 2
F 4
Q 2 4
C 0
C 3
```

Sample Output 1

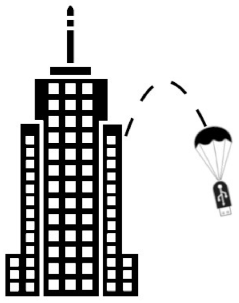
```
166 10
175 34
NO
YES
```

This example corresponds to the input illustrated above.

Constraints

- $2 \leq N \leq 2\,000\,000$.
- $1 \leq OPI_i \leq 100$.
- $1 \leq Q \leq 5000$

6 Parachuting USB



You are a penetration testing specialist with a mission of exfiltrating data from a large and secure office building. Together with your team you have already planned on how to infiltrate the building and copy the sensitive data onto a USB drive. There is strong security at the door, so it seems that the only way of extracting the USB drive from the building undetected is to throw it out of a window attached to a small parachute, to be retrieved from the street below. The sensitive data is kept on

the ground floor, which is where you have to perform the copying. There is a great risk of being caught, as you ascend the building, so you want to extract the USB as quickly as possible. For this reason, you need to know the lowest possible floor from which you can safely extract. The day before

the mission you and your team want to determine which is the lowest floor from which you can drop the drive without it getting destroyed. If the level is too small the parachute will not have enough time to open, and the drive will hit the floor at unacceptably high speed. If the level is too high, you run unnecessary risk of being caught. Luckily, you find a similar building for which the owner allows you to do your experiments, and determine the exact optimal floor. Unfortunately, they require you to pay each time you drop the USB drive, as it requires you to be walking through their offices each time, causing some disturbance.

Knowing that you have a certain amount of test drives that you can throw, what is the minimum amount of drops you need to perform to **guarantee, for any possible safe level** that you find the lowest safe level the USB drive can be dropped from.

- A USB drive that survives a test fall can be used again. Test falls that are successful do not effect the longevity of the USB
- A broken USB drive must be discarded, as it it irreparably broken
- The effect of a fall is the same for all USB drives
- If an USB drive breaks when dropped from a floor, it would have broken if it had been dropped from any lower floor
- If an USB drive survives a fall from a floor then it would survive a longer fall, form any floor above that one
- It is neither guaranteed nor ruled out the the first floor is safe or breaks the drive
- It is neither guaranteed nor ruled out the the last floor is safe or breaks the drive

What is the least number of USB drive drops that is guaranteed to work in all cases?

Input/Output

Two lines, containing integers N, the number of floors of the target office building and M the number of test USB drives at our disposal, respectively.

Output a single integer, the minimum amount of test drops necessary to know with certainty the lowest safe level from which the USB can be ejected.

Examples

Sample Input 1

```
4
1
```

Sample Output 1

```
4
```

Sample Input 2

```
36
2
```

Sample Output 2

```
8
```

Constraints

- $0 \leq N \leq 10\,000$.
- $0 \leq U \leq 10$