S.T.A.C
IIT Mandi

27/12/2020

Start Time: 0:00 hrs.

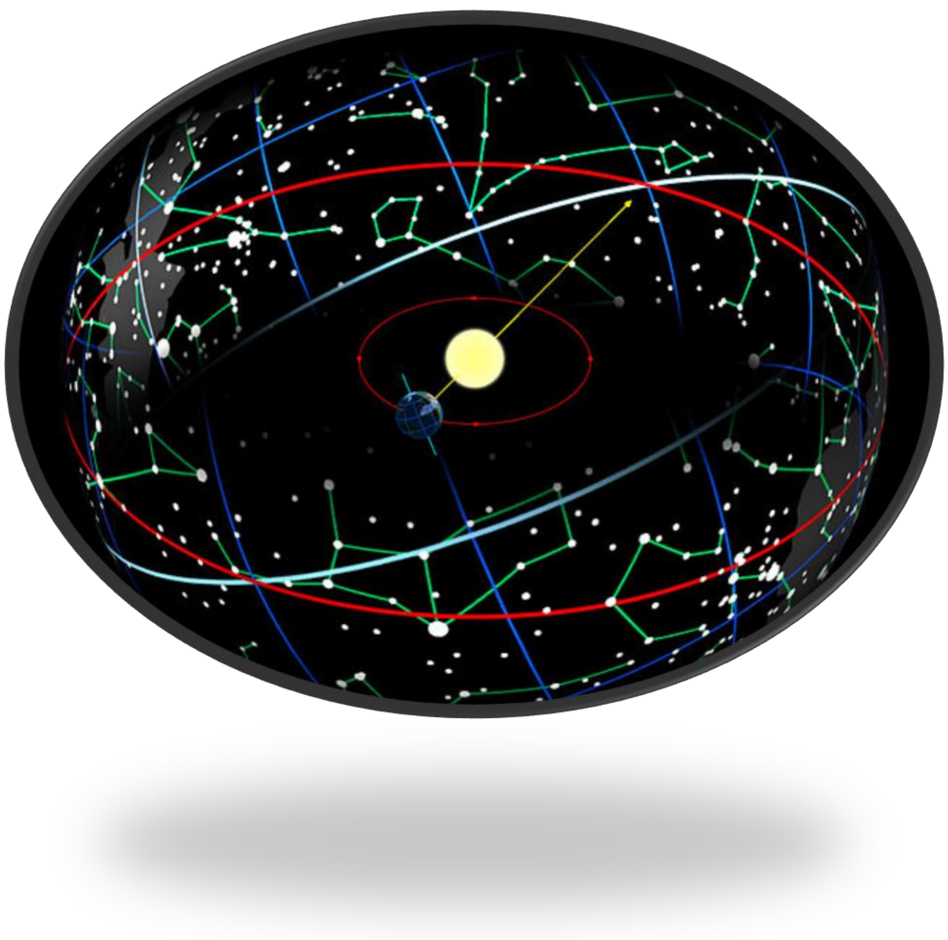
End Time: 23:59 hrs.

Total Questions: 9

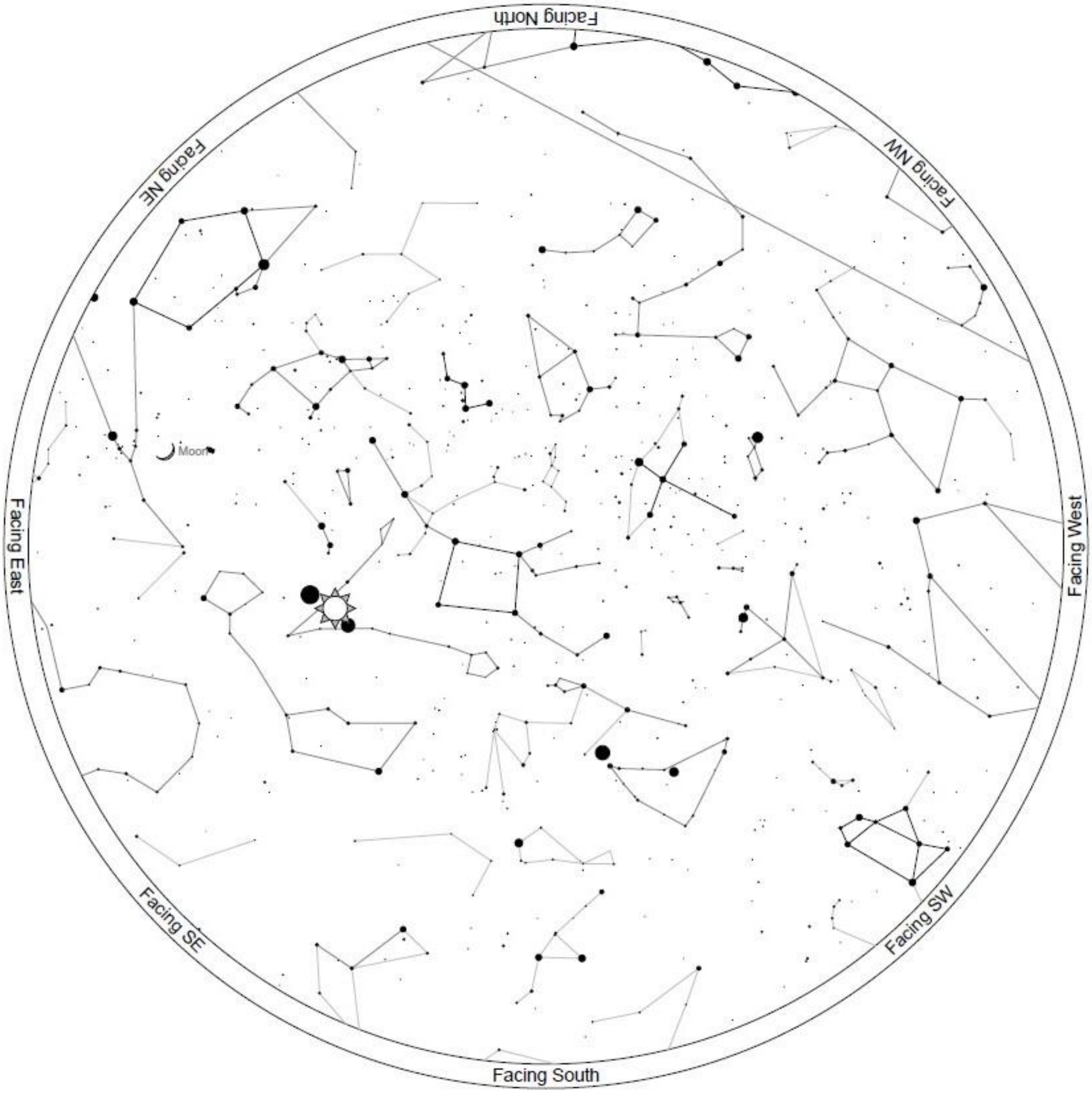
Total Tokens: 100

1. Zodiac Zest

Imagine this: take a ray that starts right at the center of our planet and stretch it straight through the center of the sun and beyond the solar system. Now as the earth revolves around the sun, this axis spins around pointing out radially towards different directions. The plane this ray traces out is the ecliptic plane. Now visualize the area within $\pm$ 8^0 of the ecliptic plane as the earth orbits the sun. This area is the zodiac (the word derives from the Ancient Greek word *zōidiakòs kýklos* meaning the "circle of little animals") and the nearest constellations to our planet that fall in this zone are the Zodiac Constellations.



Now, identify all the Zodiac Constellations in the Star Map given below.



[5 tokens]

2. Lucy *'and'* the Sky with Diamonds

- Inspired by The Beatles

Lucy is standing at the Gravity (S.T.A.C Club Room) terrace looking down at the many Diyas that have lit up the campus on Diwali (14th November, 2020). She now looks straight up to see the night sky lit up by the stars. She immediately recognizes the constellation above. She looks around and sees many more. However, since the campus is located in a valley, the mountains block the view to some of the constellations.

Help Lucy write a program that takes the following as input in the given order:

[Year, Month, Date, Hour, Minutes, Longitude, Latitude, Azimuth, Altitude]

And outputs the constellation in the direction specified by the azimuth and altitude coordinates relative to the latitude and longitude coordinates for the specified time (IST). Also output the right ascension and declination coordinates in hours and degrees respectively of the constellation.

Note: The RA/DEC of a constellation is approximated by its centroid.

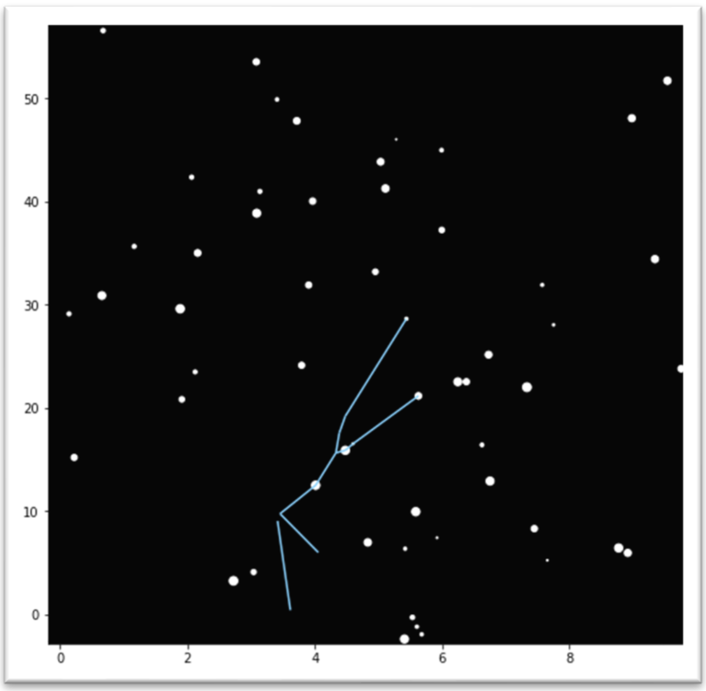
Next, find out the constellation right above Lucy and the one opposite to it (on the other side of the planet) if the time is 10:00 P.M. and name all the constellations in the following directions:

{N, E, W, NW, NE, SE} taking altitude as 0 relative to the terrace. [10 tokens]

3. Lucy, Tony and The Stars

Tony was really impressed by Lucy's program and decided to further help her make it better. He suggested that the output of the previous program can be further passed as an input to another function that then plots the constellation. This plot will have the **x-axis as the right ascension values** and the **y-axis as the declination values**. Each point will then represent the position of a star. The stars that belong to the required constellation will then be connected using constellation lines to bring out its shape.

For instance, if the constellation is Taurus, the respective output should look something like this:



Complete the Program!

[15 tokens]

4. Cool Stack

The Messier objects are a set of 110 astronomical objects catalogued by the French astronomer Charles Messier in his Catalogue des Nébuleuses et des Amas d'Étoiles ("Catalogue of Nebulae and Star Clusters"). Because Messier was only interested in finding comets, he created a list of non-comet objects that frustrated his hunt for them. The compilation of this list, in collaboration with his assistant Pierre Méchain, is known as the Messier catalogue. This catalogue of objects is one of the most famous lists of astronomical objects, and many Messier objects are still referenced by their Messier number. The catalogue includes the astronomical deep-sky objects that can easily be observed from Earth's Northern Hemisphere; many Messier objects are extremely popular targets for amateur astronomers. Here you can read more about messier objects.

Andromeda Galaxy (M51)

Mangla is frustrated because he can't see cool messier objects using STAC's telescope. He decided to make a python program which can STACK more cool images of all messier objects. You know that he is sleepy all the time, but he wants your help to make this program.

You are expected to make a script which can save all images of messier objects in a directory named CoolSTAC.

Directory structure:

```
Project
| README.txt (if any)
| main.py
|
└───CoolSTAC
|     | M1.png
|     | M2.png
|     | ...
```

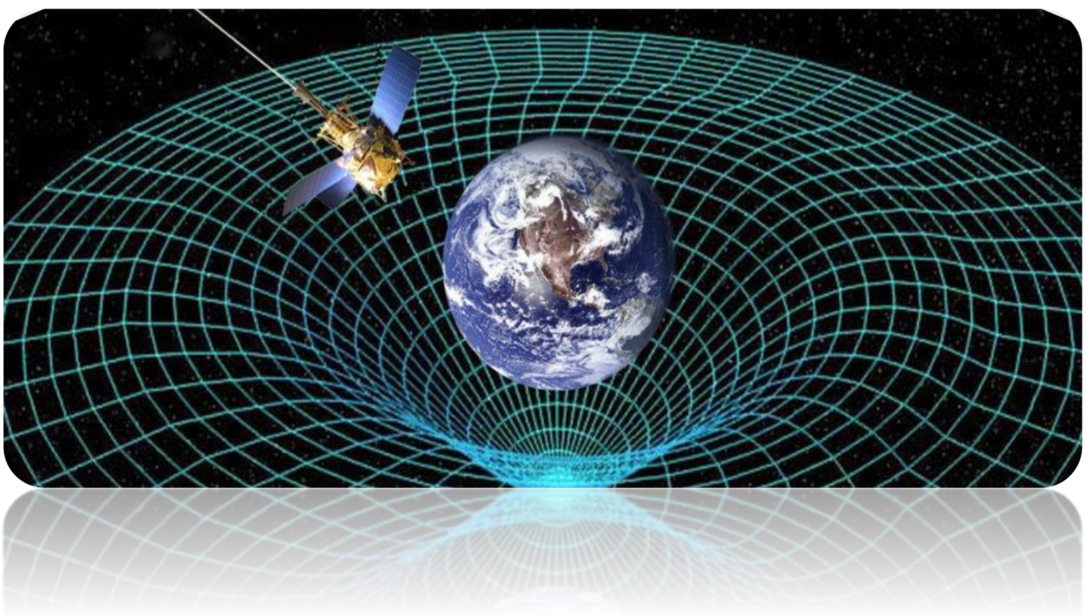
Rules:

1. The name of the image should be 'M' + '<id of messier object>' + '.png'.
2. You are NOT allowed to download data in any format, your program should fetch data itself in the form of 'fits' file.
3. You also know that Mangla doesn't know the exact number of messier objects, so you can skip up to 25 messier objects.

[10 tokens]

5. Struggle for keeping time alive!

The year is 2315… Earth has been almost completely submerged in water due to complete melting of the polar ice caps, which was followed by an ice comet strike. Luckily, this impact had been predicted, and some humans were safely transported to 2 big spaceships, The Ark, and The Messiah, which are humankind's last hope. These spaceships are in orbits around Earth, keeping Earth time currently. Can you help the last humans calculate the total time dilation between the spaceships and Earth, so they can properly set their clocks?



Problem:

Write a program to calculate the total time dilation between Earth and any of the spaceships, given the radius of its orbit.

Assumptions:

Consider the orbits to be circular. And consider the Atomic clock on Earth to be on the equator.

Input:

First line will take Q, the number of test cases.

Followed by Q lines, each will be the distance of the spaceship from Earth, in meters, in scientific notation using e or E (like say 1.2345E6).

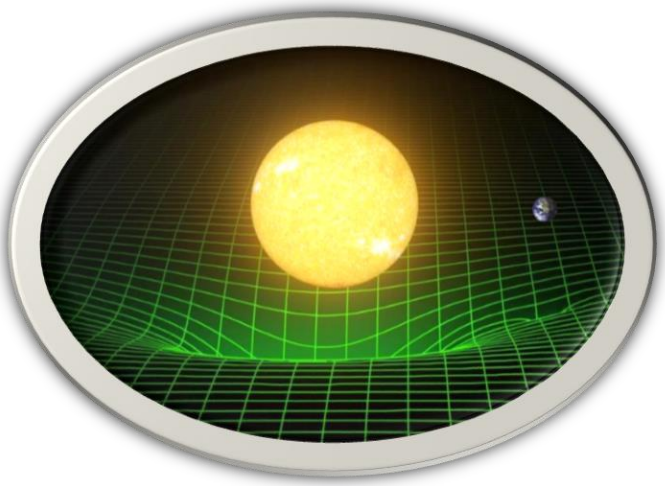
Output:

A single line after each inputted test case, containing the time dilation of the spaceship per Earth day, in seconds, in scientific notation using e or E (like 1.234E-5), with the number of decimals rounded to 5 digits.

[8 tokens]

6. For the survival of the species!

After much discussion, the remaining scientists eventually concluded that waiting for Earth to return to normal, could take very long, and they might run out of resources on the spaceships. Hence, they decided to send Messiah to a farther orbit, in the Asteroid belt or the Kuiper belt to look for the required resources. Of course, again the scientists will need to calculate the time dilation between the two spaceships in their new orbits. Could you once again, spare some time off your assignments (Yes, even in such times, profs will not spare you from assignments ☺) and help the scientists in your spaceship to keep a track of time?



Problem:

Write a program to calculate the total time dilation between the two spaceships, one of which is in Earth's orbit, and the other in Sun's orbit.

Assumptions:

Consider the orbits to be circular. Also, consider the orbits of Earth, The Ark and The Messiah to be coplanar. Average out the velocity components of the Ark over a long period, to get the results independent of the relative positions of the two spaceships. Neglect the gravitational influence of the celestial bodies on the spaceships, other than those around which they are revolving (i.e. only consider Earth's gravitational influence on the Ark, and only Sun's gravitational influence on the Messiah).

Input:

First line will take Q, the number of test cases.

Followed by Q lines, containing first the distance of The Ark from Earth, and then the distance of The Messiah from Sun, the values will be space-separated, in meters, in scientific notation using e or E (like say 1.2345E6<space>3.1415E9).

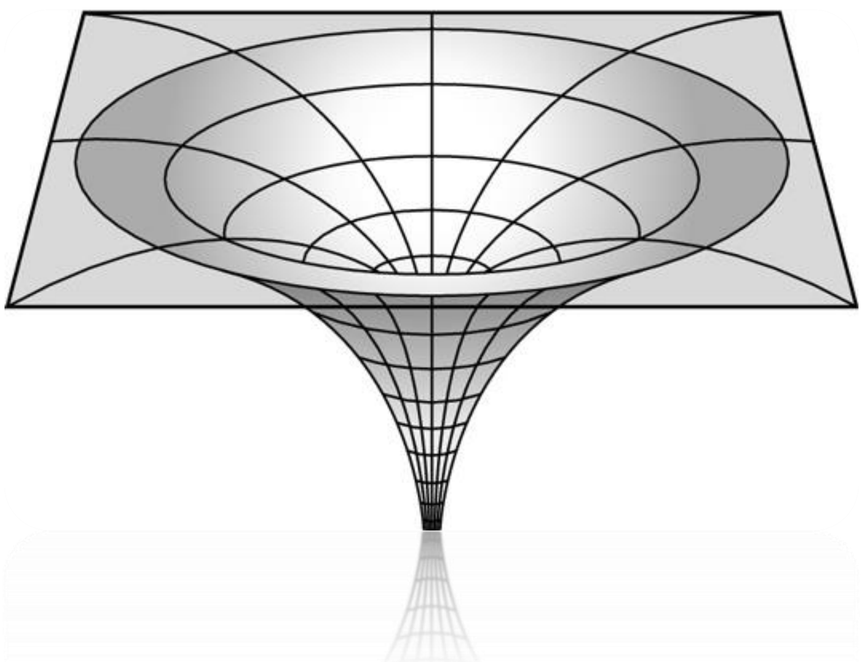
Output:

A single line after each inputted test case, containing the time dilation of the Messiah per Earth day on the Ark, in seconds, in scientific notation using e or E (like 1.234E-5), with the number of decimals rounded to 5 digits.

[22 tokens]

STRING THEORY AND BLACK HOLES

String theory is a theoretical framework in which the point-like particles of particle physics are replaced by one-dimensional objects called strings. String theory describes how these strings propagate through space and interact with each other. On distance scales larger than the string scale, a string looks just like an ordinary particle, with its mass, charge, and other properties determined by the vibrational state of the string. The essence of String theory is in study of Black holes. You can get more information regarding black holes and its entropy calculation by string theory in the paper *Microscopic origin of the Bekenstein-Hawking entropy* (1996) by Andrew Strominger and Cumrun Vafa.



7. MARS EXPLOSION

Assume Mars explodes one day and changes to a Black hole. Calculate the entropy generated by that black hole. Mention all the assumptions that you take for the calculations. Assume the radius of the event horizon is equal to 10^5 times the radius of any massive object that should be compressed to create a black hole.

[5 tokens]

8. MATHEMATICALLY EXPLAINING A PLANETS' FORMATION

Assume a planet which changes its area but we don't know its relation (area may both increase and decrease). We have the values of the area for about 100000 timestamps. We don't know the underlying distribution of the data. But we can safely assume it to be a linear superposition of multiple Gaussians. The data is provided at this link.

Constraints:

The areas are integers ranging from 1 to 1000000.

For this problem, consider the area of the event horizon to be as the same as the area of the planet.

Sub-parts:

1. Find the most appropriate value of Q for the number of Gaussians. Find their corresponding mean and standard deviation.
2. What does Q signify?
3. Consider one subset of these values of area. Prove that there exists a subset whose sum is 99999.

[5+2+3 tokens]

9. NASA'S ROVER ON AN ICY PLANET

In the above problems, we discussed the formation and destruction of the planet only ☺. Now, the planet is formed. After getting this information, NASA is planning to send a space rover to that planet to know more about it. They also know that the planet has heavy snowfall, which is quite weird. However, this snow will make the travelling of the rover difficult.

So, now we have to design a program which will make the rover move on the surface of the planet. First, we have to assume that the surface of the planet is a 2D plane. Thus, its surface is grid with h rows and w columns. Rows are numbered 1 through h from top to bottom. Columns are numbered 1 through w from left to right. Every cell is either allowed (denoted by '.' in the input) or permanently blocked (denoted by '#'). Every day a few allowed cells are temporarily blocked by snow due to heavy snowfall as mentioned above. Note, that this block works only on this particular day and next day any of these cells might be allowed again (unless there is another temporary block).

It's possible to move directly between two cells only if they share a side and none of them is permanently or temporarily blocked. NASA plans to land the rover at the top left cell, while its destination is at the bottom right cell. Every day the rover should first go from its starting point to the destination and then return back to its starting point. Since the rover can't store the path on where it is moving, it doesn't visit the same cell twice in one day, except for the cell with his starting point, where he starts and ends. If it can reach a destination and return back safely avoiding revisiting cells, it marks its daily target successful.

There are q days you must process, one after another. For each of these days you should check if its target is successful and print "YES" or "NO" on a separate line. In order to be able to read the description of the next day you should print the answer for the previous one and flush the output.

It's guaranteed that a day with no cells temporarily blocked by snow would be interesting. It's also guaranteed that cells with the rover's starting point and destination are never blocked (neither permanently nor temporarily).

Input:

The first line of the input contains three integers h, w and q ($2 \le h, w \le 1000$, $1 \le q \le 10\,000$) — the height and the width of the grid, and the number of days, respectively.

Next h lines describe which cells are allowed and which permanently blocked. The i-th line contains a string of length w, describing the i-th row. Every character is either '.' (denoting an allowed cell) or '#' (denoting a permanently blocked cell). It's guaranteed that a day with no cells temporarily blocked by snow would be interesting.

Then, the description of q days is given. The description of the i-th day starts with a line containing a single integer ki ($1 \le ki \le 10$) — the number of cells that are temporarily blocked by snow on that day. Each of next ki lines contains two integers ri, j and ci, j ($1 \le ri, j \le h$, $1 \le ci, j \le w$), representing a cell at the intersection of the row ri, j and the column ci, j. The given ki cells are distinct and none of them is permanently blocked. Also, none of them contains the starting point or the destination.

Output:

For each of q days print "YES" if that day is successful, and otherwise print "NO", both without the quotes. After printing an answer, you have to both print the end-of-line character and flush the output. Then you can proceed to the next day. You can get Idleness Limit Exceeded if you don't print anything or if you forget to flush the output.

[15 tokens]