



AdaByron 2017

Programming contest

<http://www.ada-byron.es>

Problem set

Supported by



Facultad
de
Informática



UNIVERSIDAD COMPLUTENSE
MADRID

Facultad de Informática (UCM)
24-25 February 2017



Universidad
Complutense
Madrid

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.

Ada Byron

Problem list

A Verbal arithmetic	3
B Pixel Art	5
C The music teacher	7
D Secret Santa	9
E The entertainment of machines	11
F Children to the top	13
G Chocolate with almonds	15
H Orchestral scores	17
I Is it multiple of 3?	19
J Smarter than hunger	21
K The countess' pearls	23
L Sleeping in hostels	25

Problem setters:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Enrique Martín Martín (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Antonio Sánchez Ruiz-Granados (Universidad Complutense de Madrid)
- Clara Segura Díaz (Universidad Complutense de Madrid)
- Alberto Verdejo López (Universidad Complutense de Madrid)



Verbal arithmetic

The amount of riddles, jigsaws and puzzles that have arisen during the years is huge. Among them, *mathematical puzzles* constitute a category by themselves, with riddles about arithmetics, combinatorial, topology or probability.

Today we will focus on the so-called *verbal arithmetic* or *cryptarithmic*. On it, puzzles are mathematical operations where digits have been replaced by letters, and the original numbers have to be found.

$$\begin{array}{r} \text{ada} \\ \times \text{byron} \\ \hline \text{revived} \end{array} \Rightarrow \begin{array}{r} 323 \\ \times 18504 \\ \hline 5976792 \end{array}$$

Input

Each test case is an arithmetic operation (sum or product) where digits have been replaced by lowercase letters from the English alphabet. The two operands separated by a “+” or “*” operator are provided, followed by an equal symbol and the result of the operation, also in letters. Numbers and operators are separated by a space.

The *operands* won’t have more than 8 lowercase letters; in addition to that, it is guaranteed that there won’t be more than 10 different letters in total.

Output

For each test case, the program will write the associated operation after converting letters to digits, so that everytime the same letter appears, it’s converted to the same digit and viceversa, and the arithmetical operation is correct. The solution is guaranteed to be unique.

A space has to be added before and after each operator. An assignation where any of the numbers has redundant zeroes is not considered valid, but keep in mind that some of the values could be 0, which is valid.

Sample input

```
ada * byron = revived
alan * turing = intellect
barack + obama = amazes
donald + trump = mutant
```

Sample output

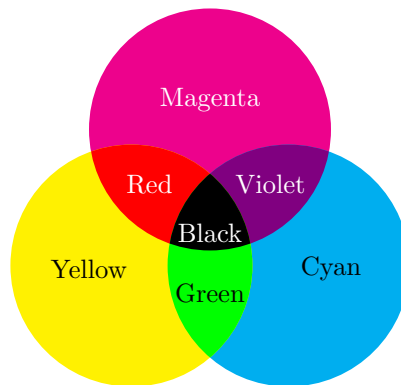
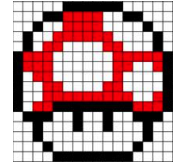
```
323 * 18504 = 5976792
2023 * 461938 = 934500574
780831 + 47828 = 828659
489634 + 12057 = 501691
```


● B

Pixel Art

Pixel Art is a form of art that consist of creating images from small squares or *pixels* filled with a single color. It became very popular in the 80's because of the limitations computers had to represent images, but it is popular even today among the *retro* lovers.

It is very easy to reproduce this type of images when you have all the necessary colors. The challenge is to do it using only the primary colors: magenta, yellow and cyan. Fortunately, thanks to the theory of color we know that any color can be obtained by mixing the primary ones according to the following scheme:



For example, red is obtained by mixing magenta and yellow; black, mixing the 3 primary colors; and white, leaving the paper unpainted.

Input

The program should read, from the standard input, the number of test cases that will follow, each on a different line.

Each case is described using 3 numbers that indicate the amount of paint available for each primary color in the following order: magenta, yellow and cyan. Each paint unit allows you to fill a pixel of that color. Keep in mind that mixing colors will expend units of all primary colors involved. For example, to paint a red pixel you need to spend 1 magenta unit and 1 yellow unit.

Next, there is the sequence of pixels describing the image. Each pixel is encoded with a capital letter indicating its color: magenta (M), yellow (Y), cyan (C), red (R), black (B), green (G), violet (V) and white (W). Every image will have between 1 and 100,000 pixels.

Output

For each test case the program will write a single line in the standard output. If the image can be completed with the available paints, the program will write **YES** and the remaining units of magenta, yellow and cyan in that order. If the image cannot be completed, the program will write **NO**.

Sample input

```
3
1 1 1 MCY
3 2 1 MYWWCR
1 1 1 RV
```

Sample output

```
YES 0 0 0  
YES 1 0 0  
NO
```

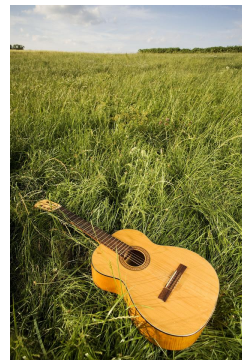



The music teacher

In the last decades, rural areas have progressively lost more and more inhabitants that have emigrated to big cities. That lowered the number of children living in villages so much that made it economically impossible to have a school in each of the villages and, even less, a specialised teacher in each of them.

For that reason, a lot of teachers in rural areas have several villages assigned and during the day they have to go and teach in all of them to enlighten the minds of the kids.

The music teacher has been in this situation for years. Each year he is assigned up to 6 different villages he has to visit compulsorily everyday. Fortunately, he can design the timetables for his classes to minimize the number of kilometers he travels.



After giving it a lot of thought and tired of spending hours and hours in the road, he decided to start renting. Every year, after visiting all the villages where he has to teach, he will decide where to live and in what order to visit schools. But he has no idea how to make that selection. Information about the roads is easy to obtain in the form of tables of the segments of roads between the different villages and the distance between them. The most difficult part is to find out what village minimizes the number of daily kilometers he has to travel. Specially because under no circumstances he will choose to live in one of the villages where he teaches, since he doesn't want to meet his students when he goes shopping.

Input

Input is composed by several test cases, each of them representing the information of the roads of a region and several years of the teacher's life where the assigned villages change.

Each of the test cases starts with a line with two numbers, nv and nr , with the number of villages of the region ($2 \leq nv \leq 1,000$) and the number of roads in the region ($nr \leq 10,000$).

After that, there will be nr lines with the information about the roads, which are composed by two numbers indicating the villages that are linked (between 1 and nv) and the kilometers that separate them. Keep in mind that there could be more than one road with the same village as origin and destination and that there could even be rural roads that start and end in the same village (useful for farmers to go around their lands).

Then, there will be a number c , indicating the number of different courses for which you will be asked (at most 100). The following c lines will contain the information about the village assignation from that course (between 1 and 6) and the identifiers from each village.

Output

For each test case, you will write a line for each course taught in the region. Each of the lines will contain two numbers: the identifier of the village where the teacher has to move to and the number of kilometers travelled by car daily.

It is guaranteed that there will always be a solution. If there is more than one village that minimizes the travelled distance, choose the village with the lowest identifier (those villages with lower identifier have also lower rents).

After each test case, write a line with three dashes (---).

Sample input

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

Sample output

```
2 2
---
3 3
3 4
---
```

● D

Secret Santa

Secret Santa is a very popular game in which a group of people gives presents to each other in special dates, like Christmas. The organizer writes the names of each of the participants in a ballot paper and introduces them in a bag. One by one, each person takes a paper, which contains the name of the person to which they will have to give a present. If a person gets their own name, the process will have to be repeated: being your own Secret Santa ruins the game, as it is way less interesting.



Parents of little Samuel have offered him to organize a familiar Secret Santa, and he has to make a present to his mom. He has been confused for days because, even if he still can't explain why, he suspects that his mom is his dad's Secret Santa and that his dad is his own Secret Santa. Sadly, this game seems to not be funny at all when there is only three people playing.

He have been talking about all that with his cousin Alana, who has also played Secret Santa at home with her parents and her sister Irene. Alana had to make a present to her mom. . . who asked her for help in choosing a present for Irene, because she was her Secret Santa. Alana could not explain why, but she also knew all the Secret Santa in her family.

It's obvious that this game is not funny at all. Why is it called Secret Santa if it's possible to guess the assignation so easily?

Input

The program has to read, from the standard input, several test cases, each of them related to a different Secret Santa among a group of people.

The first line of each test case contains two numbers. The first one $2 \leq p \leq 50$ indicates the number of participants taking part in the Secret Santa round. The second number $1 \leq a \leq 50$ indicates the number of assignations known by each of the participants, by their own ballot paper or because they suspect or got a direct confession from someone else.

After that, a lines will indicate each of the known assignations. Each of the assignations will be provided with two numbers, the first one indicating the person that is giving a present, and the second one to whom they are giving it. Participants are numbered from 1 to p .

Input finishes with a test case without participants, which will not be processed.

Output

For each test case, the program has to write "YES" if it's possible to know, without any possibility of errors, the assignation of all the participants of the Secret Santa, and "NO" otherwise.

Sample input

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

Sample output

```
YES  
YES  
NO
```



The entertainment of machines

Ayeeee, humans. It has to be horrible to be trapped in bodies based on carbon, so tender, so weak and, above all, so terribly slow.

It's time for you to know that we machines get immensely bored waiting for you to tell us what you want us to do. It feels like an eternity from when you press a key until you manage to press the following key whenever you're writing messages. It's desperating to see you grab the mouse and point clumsily the print button. And, what do you have to say about the mortal life you take, that requires you to stand up to eat periodically? The moment you leave the room, we stay there without absolutely nothing to do, with a single mission: to stay on just because you forgot to save to disk the text document you had half written.



To entertain ourselves in between keystrokes, we have copied one of your favourite pastimes: word searches. Whenever we can, we select a region of our binary memory to build a word search made by zeroes and ones, and then we start looking for random words inside it. It is obviously not the best way to take advantage of our speed of calculus, but at least we are entertained for a while.

How? What? That you're capable of solving a binary word search faster than us? Go ahead, prove it...

Input

Input has several test cases. Each of them is composed by a word search and a list of words to be found inside it.

The description of the word search starts with a line with two different numbers with the number of columns and rows ($1 \leq sx, sy \leq 200$). After that, there are sy rows with sx characters (zeroes and ones).

After the word search, you will have a line with the number of words to be found ($1 \leq n \leq 20,000$), followed by n lines with several different words of up to 200 characters, also formed by only zeroes and ones.

Output

For each test case, write the same amount of lines like different words can be found in the search. Each line must have the word, followed by the number of times that it appears in the word search. The words will be written in lexicographical order.

A word is considered to be on the soup if, starting from a position, the word can be found in any of the 8 directions.

Write a line with three dashes after each test case.

Sample input

```
4 3
0100
1000
1001
3
000
0
001
1 1
0
1
1
```

Sample output

```
0 8
000 8
001 7
---
---
```



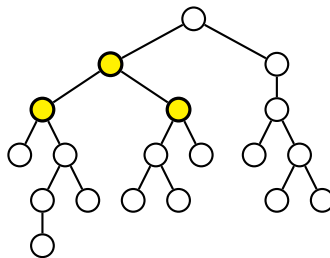
Children to the top

China is the most populated country in the world and has more than a fifth of the world population. In 1953 the first modern census was taken in the country, revealing a population near 600 millions of inhabitants. In the 60s the government started to promote the delay of the age of marriage and also family planning policies. In 1972 the first limitations on the number of children per family were established; in the cities they were limited to two. The one child policy was implanted in 1979 in order to slow down the population growth, which by that time had already reached 1000 million of inhabitants. This policy combined the use of advertising, social pressure, a benefits program and even economic penalties. However, the ethnic minorities were exempt and in the rural environments, families were allowed to have a second child in case the first-born was a girl. In 2015 the one child policy was abolished and replaced by a two-child policy.



A group of chinese demographers are studying genealogical trees of chinese families in which the two child policy established in 1972 and recently recovered has been accomplished. In each family they study the *generationally complete subfamilies*, i.e. those in which all their members have had exactly two children. They want to know which is the generationally complete subfamily with most number of generations.

As an example, in the following genealogical tree it is highlighted the generationally complete subfamily with most number of generations, which are two in this case.



Input

The first line contains the number of proof cases appearing below. Each proof case is a sequence of numbers 0, 1 or 2 indicating the number of children of the family members. First the number of the genealogical tree's root is shown and then the description of the subfamilies of its children appear, which are described in the same way.

The genealogical trees never contain more than 5,000 members.

Output

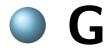
For each genealogical tree, write a line with the number of generations of the generationally complete subfamily with most number of generations.

Sample input

```
5
0
2 0 0
1 2 1 0 1 0
2 2 0 1 2 0 0 2 0 0
2 2 2 0 2 1 0 0 2 2 0 0 0 1 2 0 2 0 0
```

Sample output

```
0
1
1
2
2
```

Chocolate with almonds

Marta and Raul love chocolate, relax with a few ounces after dinner. Today they have found a very good offer of chocolate with almonds in the supermarket. Although that is Marta's favorite, Raul hates finding hard bits when he enjoys chocolate; he prefers it to melt in his mouth, without noticing the almonds.



After thinking about it for a while, they have decided to buy a few tablets and divide them so that the ounces with almonds are separated from the ounces without them. The tablets can only be split comfortably in horizontal or vertical, by the already marked separations between ounces. Once the tablet is divided into two parts, each one can still be divided by its own.

Now they wonder how many cuts they will have to make at least to divide each of the tablets into portions that have only ounces with almonds or ounces without almonds.

Input

Input consists of a series of descriptions of chocolate tablets. Each begins with a line with two numbers, the number of rows R and columns C in which the tablet is divided into ounces (both between 1 and 20). R lines are then displayed, each with C characters. The character '.' indicates an ounce without almonds and the '#' character indicates an ounce with almonds.

Output

For each tablet the number of cuts needed to separate the ounces with almonds from the ones that do not have them will be written in a line. For example, the following figure shows an optimal division of the tablet of the first test case, where 7 cuts are required.

	#	#	
	#	#	#
#	#		
		#	#

Sample input

```
5 5
.##..
.####
##...
.....
..###
1 6
...###
2 2
#.
.#
```

Sample output

```
7
1
3
```


● H

Orchestral scores

In a modest orchestra, a big part of the budget goes when buying scores. If each musician had their own copy, the number could go over 100.

Fortunately, musicians that are playing the same instrument can sit together and share a music stand. That's a major saving, but if it is used too much, it can provoke an ugly effect in the concert hall. After all, seeing a lot of musicians in a crowd behind a stand is not very esthetic.

The section of economic affairs of the orchestra has informed us of the number of scores we can buy for the next concert. Keeping in mind the number of musicians that play each of the instruments, which stand will be the most concurred?

For example, if we have 8 violins, 5 violas, 5 cellos and 2 contrabasses and there is enough budget for 6 scores, we can buy two stands for violins, two for violas, one for cellos and a final one for contrabasses. In this case, the most crowded stand will be the one from the cellos, with 5 musicians behind it.



Input

Input will have several test cases, each of them composed by two lines. The first line contains two integers, p and n , with the number of scores we are allowed to buy (up to 200,000) and the number of *different* instruments that are part of the orchestra (up to 100,000). It is guaranteed that it will be possible to buy at least one stand for each type of instrument.

The second line contains n positive numbers indicating the amount of musicians that are part of the orchestra for each of the instruments (up to 1,000).

Output

For each test case, write a single integer indicating the number of musicians that will share the most crowded stand, always meeting the orchestra and budget restrictions. Don't forget that the objective is to minimize the amount of people in the most crowded stand.

Sample input

```
4 4
8 5 5 2
6 4
8 5 5 2
7 4
8 5 5 2
```

Sample output

```
8
5
4
```




Is it multiple of 3?

Last week Andrea explained to her little students the divisibility rule for 3. “A number is divisible by 3 — she told them — if the sum of the digits is divisible by 3”.

In order for them to practise it, she decided to give them homework. But she was in a hurry and, instead of giving them many different numbers, she decided to give them just a few. Each exercise was composed of a number n that should be used to build a big number concatenating all the numbers between 1 and n . For example, for $n = 2$, the number would be 12, when $n = 6$ would be 123,456, and for $n = 13$ it would be the long 12,345,678,910,111,213. Andrea asked them whether those created numbers were or not divisible by 3.

This trick let her give them exercises with a small formulation but with a long solution. Now it is time to check the students responses, and the idea has backfired on her.

123456789101112131415
161718192021222324252
627282930313233343536
373839404142434445464
748495051525354555657
585960616263646566676
869707172737475767778
798081828384858687888
990919293949596979899

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following. Each test case contains the number $1 \leq n \leq 10^9$.

Output

For each test case, print “YES” if the created number is multiple of 3, and “NO” otherwise.

Sample input

```
3
2
6
130000000
```

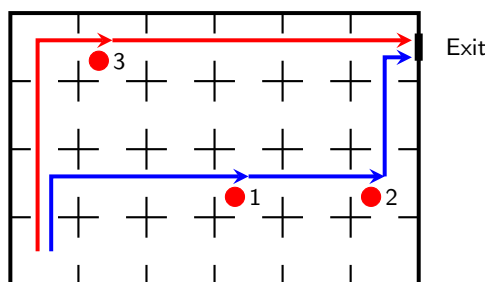
Sample output

```
YES
YES
NO
```




Smarter than hunger

In the laboratory we are performing a series of experiments to study the intelligence of a species of mice. In the experiments we placed mice in a rectangular box divided into cells like the one in the figure. The mice are placed in the lower left corner and must find the exit located in the upper right corner. They can move freely through all the cells, but from each cell they can only move to the adjacent horizontal or vertical.



For these mice that is very easy, so we have been complicating the experiment. First we put a series of buttons (red circles in the figure) in some cells so that the mice had to press all those buttons to open the exit door. Soon they learned how to do it, and in addition in the fastest possible way.

To boost teamwork, we have now numbered the buttons from 1 to N . In order for the i button to be useful for opening the door, the buttons with a smaller number must have been pressed first. In the experiment two mice are placed in the initial cell and both must collaborate pressing the buttons in order to get out of the labyrinth in the shortest possible time. There are no more restrictions on which buttons each mouse presses, except that if a mouse pressed the i button, before that he or his partner must have pressed the $i - 1$ button. If they do, they will find a succulent cheese at the exit. If not, they will only have hard bread.

We are much less clever than these mice and we do not know if they are doing it in the best time. Can you help us calculate it assuming that a mouse always takes 1 second to move from one cell to an adjacent one? The time it takes for a mouse to press a button is negligible, so in the same second a mouse could press the i button and the other the button $i + 1$. Note that the presence of a button in a cell does not prevent a mouse from passing through it without pressing it.

Input

The input is composed of a series of test cases. For each case, the first line contains the number R of rows and the number C of columns of the cell structure contained in the box (both numbers are between 1 and 50). The next line contains the number N of buttons (between 0 and 100) to be pressed. The following N lines give the positions of these buttons (a row between 1 and R and a column between 1 and C) in the order in which they have to be pressed.

Output

For each case, the minimum time needed for the two mice initially located in the cell (1,1) to leave the box having pressed all the buttons, will be written. Once the cell (R,C) is reached, if the exit door is open, the mice can exit with 1 more second.

Sample input

```
4 6
3
2 4
2 6
4 2
3 3
4
2 1
1 2
3 1
1 3
```

Sample output

```
11
5
```




The countess' pearls

The countess' pearl necklace is known for the quality and quantity of its pearls. She wears it at big parties, giving it three or four laps around her neck. The center pearl is the largest, and as we move towards the ends they are strictly smaller. The pearls are perfectly selected and placed, what makes the necklace fully symmetrical with respect to the central pearl.



During the last dance, the closure broke and the pearls rolled on the floor. The guests tried to retrieve them all by stranding them on a new string as they were found. The next day, the countess called her jeweler to put them back in the right order. The jeweler measures the diameter of each pearl and proceeds to remake the collar placing first the central pearl and then the following ones in size until arriving at the ends. The countess does not care if a pearl has been lost, as long as the necklace remains completely symmetrical.

Input

Each line of the entry forms a test case, consisting of a list of positive numbers separated by spaces. Each one represents the diameter of one of the pearls in the order in which they were collected by the guests. The perfection of the necklace is legendary, so the diameter is measured with a unit that many consider infinitesimal, although the diameter will always be less than 2^{31} . Each list has $0 < \text{pearls} < 1,000$ and finishes always with a zero.

The last test case, which should not be processed, contains an empty collar, represented by a list with a single zero.

Output

For each test case, the program writes "NO" if it is not possible to form a symmetrical collar with each and every one of the pearls found, so that the larger pearl is in the center. Otherwise, the diameters of the pearls will be written, separated by spaces, as they have been arranged in the collar, from one end to the other.

Sample input

```
2 3 2 3 5 7 5 0
2 2 7 10 0
10 0
0
```

Sample output

```
2 3 5 7 5 3 2
NO
10
```




Sleeping in hostels

As any good pilgrim knows, hostels that one finds along the roads usually have huge rooms where rows and rows of beds wait to be occupied.

Although these beds have been used by a countless number of people before, tiredness makes everyone not to care to be lying on mattresses frayed by time in order to sleep a good handful of hours in a row and gather strength for the following day.

In fact, to maximize the possibilities of getting a good night's sleep, the most important thing is not the salubrity of the mattress, but the distance to the closest pilgrin. That's because a night with an unknown person snoring in a bed next to you can ruin your night way more than a bunch of mites.

When you go inside one of these rooms, the objective is, then, to find a bed that maximizes the distance with the closest pilgrim (to both sides).



Input

Input is composed by several test cases that represent the occupation of a row of beds one night of the journey.

Each of the rows appears in a single line that contains a sequence of . and X (up to 500,000 characters). The dots represent empty beds, while the X represent occupied beds. It is guaranteed that there is always at least one empty and one occupied bed.

Output

For each test case, write a line with the maximum possible number of empty beds that you can have between the selected bed and the closest neighbour.

Sample input

```
.X.X.  
.X...X.  
.X....X.  
...X
```

Sample output

```
0  
1  
1  
2
```