Problem A Crystal Ball Factory

The Astrologically Clairvoyant Manufacturers (ACM), a pioneer in future-predicting technology, just landed a contract to manufacture crystal balls for weather forecasters around the world. Every week, a variable number of crystal balls needs to be delivered; the required amount for each week is specified in the contract.

Crystal balls are made from the highest-quality crystal, whose price fluctuates from week to week. Fortunately, the ACM is able to foresee the price of crystal for the coming weeks, thanks to its own future-predicting technology.

When the price is low, the ACM would like to buy crystal and manufacture crystal balls, storing any excess in their warehouse. On the other hand, in weeks for which the price is high, ACM would rather use the crystal balls stored in the warehouse to satisfy the demand specified in their contract. However, since there is a also a fixed weekly cost to store each crystal ball in the warehouse, and an initial cost for turning on the manufacturing machines and producing a non-zero quantity of crystal balls, the decision is not always simple.

Can you help them fulfill their contract at minimal cost?

The first line of each test case (representing a contract) will contain the number of weeks for which the contract will last. The next line will contain the non-negative integers b, k and n, where b is the base cost for manufacturing a non-zero quantity of crystal balls on a given week, k is the cost for storing each crystal ball in the warehouse for a week, and n is the maximum capacity of the warehouse.

The following lines will describe the weeks specified in the contract in chronological order. Each week is described by a single line which will contain a pair of non-negative integers c and r, where c is the cost for manufacturing a new crystal ball using new crystal bought this week, and r is the number of crystal balls which must be delivered this week. A crystal ball can be manufactured and delivered in the same week if appropriate, in which case it won't need to be stored in the warehouse at all.

The last line of the input will contain the integer 0 and should not be processed.

Problem A Crystal Ball Factory

For each test case, output the minimum amount which the ACM will have to spend in order to fulfill the entire contract.

All the numbers in the input will be at most 1000.

sample input:	**sample output:**
4 1 0 1000 1 1 12 4 1 0 1000 1000 2 0 100 1 1 1000 1000 101 0	1007 101101

Problem B Sudoku

A Sudoku puzzle, once solved, is a 9x9 grid of digits organized as a 3x3 grid of smaller 3x3 units. Each of the nine rows must contain every positive digits exactly once, as do each column and also each 3x3 unit. The puzzle is to start from a partially filled 9x9 grid and to fill in the remaining cells using only logic. The puzzle maker usually makes sure that the solution will be unique and that it can be reached using deduction only, without guessing.

		4			9			8
	3			5			1	
7			4			2		
3			8			1		
	5						9	
		6			1			2
		8			3			1
	2			4			5	
6			1			7		

This number placing game is gaining popularity in the west, and every second newspaper publishes weekly instances of the puzzle. Somewhere at the head of one such newspaper, someone decided that buying individual instances from a puzzle maker would be too expansive, and instead decided to steal puzzles from other newspapers and also to print randomly generated Sudoku-like grids.

One week later, his assistant gets stuck with the job of printing the solution to the Sudoku puzzles his boss previously published. Unfortunately, his boss doesn't have those solutions, the randomly generated problems don't have *any* solution, and he doesn't even remember which is which. In despair, the assistant calls for your help.

The first line of the input will contain the number of test cases. Each test case will consist of a 9 by 9 grid of characters, where each character will either be '?' or a digit between 1 and 9 inclusively.

For each test case, you must print back the grid to the standard input, replacing each question mark with an appropriate digit to solve the Sudoku. If a test case does not allow any solution, output "impossible" instead of a completed grid. If a test case do allow a solution, you can assume that the solution will be unique, and that theoretically it could be reached without guessing.

Test cases are separated by "---" both in the input and in the output.

Problem B Sudoku

sample input:

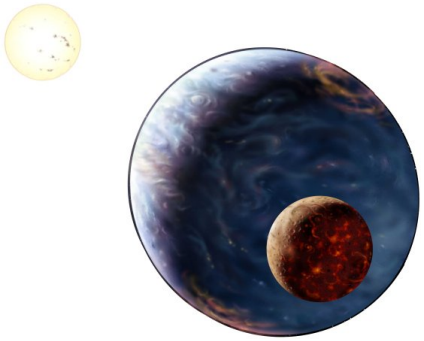
```
3
??4??9??8
?3??5??1?
7??4??2??
3??8??1??
?5?????9?
??6??1??2
??8??3??1
?2??4??5?
6??1??7??
---
??4??9??8
?3??5??1?
7??4??2??
3??8??1??
?5?????9?
??6??1??2
??8??3??1
62??4??5?
6??1??7??
---
3?1????76
7??9?????
2?5?3????
?????64?1
???2?1???
1?25?????
????8?9?3
?????9??4
51????7?8
```

sample output:

```
264319578
839257416
715486239
372894165
451632897
986571342
548763921
127948653
693125784
---
impossible
---
391452876
764918532
285637149
953876421
678241395
142593687
426785913
837169254
519324768
```

Problem C Planet Alignment

The Astrologically Clairvoyant Manufacturers have always used the celestial skies to make more accurate forecasts. Most of their predictions are based on a Star-Planet-Satellite alignment. There are some scientific reasons for this, mostly tidal changes: the gravitational pull of these celestial bodies will pull the water towards them. The greatest tides occur when the Sun, Earth, and Moon are perfectly aligned.



Always willing to expand its share of the market, the ACM has decided to generalize its predicitions to other planetary systems. However, on those systems the celestial bodies move at different speeds than in ours, so the predictions which depend on the Star-Planet-Satellite alignment must be recalculated. Can you help them automate some of their computations?

For some stellar system, at some point of the future denoted as $t = 0$, the star, a planet and a moon of the planet will be aligned on the galactic x axis. The star will be at position $(0, 0)$, the planet will be at position $(p, 0)$, and its planet will be at position $(m, 0)$. The planet moves around the star in a perfect circle, on the galactic xy plane, and completes a revolution in u Earth days. Similarly, the moon revolves around the planet in v Earth days, in a perfect circle and on the same galactic plane. In other words, at $t = u$ the planet will be back at position $(p, 0)$, and at $t = v$ the moon will once again have the same y coordinate as the planet and will be on the same side of the planet as when t was zero.

When will the three celestial bodies be aligned again?

The first line of input will contain a non-negative integer n, which represent the number of test cases. For each test case, you will be given one line containing the non-zero integers p, m, u and v, all of which could be negative. The distance between the moon and the planet will be strictly smaller than distance between the planet and the star, and the revolution durations won't cause the celestial bodies to be permanently aligned. A positive revolution duration indicates that a celestial body revolves counter-clockwise, and a negative revolution duration indicates clockwise motion.

For each test case, find the minimum positive number t such that after that number of Earth days, the three celestial bodies will be perfectly aligned again. Output that number with three decimal places, rounding to the nearest allowed value.

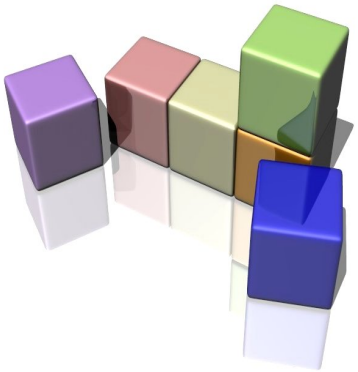
Problem C Planet Alignment

The magnitude of the numbers in the input will be at most 1000.

sample input:	sample output:
2 1 -1 2 1	0.500 1.000

Problem D

Sum of Cubes



According to Goldbach's conjecture, every even number can be expressed as a sum of two odd primes. Which numbers can be expressed as the sum of two cubes?

For each test case, a line will contain a positive integer n which will be at most one million. For each test case, output two integers x and y such that $x^3 + y^3 = n$. If there are many such pairs of numbers, choose x to be as small as possible. If there are no such pairs, output "impossible".

The last line will contain the integer 0, which should not be processed.

sample input:	**sample output:**
1	0 1
2	1 1
3	impossible
1000000	0 100
0	

Problem E Bold ASCII Lines

Your grandmother recently developed a passion for the internet and the new technologies in general, and is now discovering the wonders of ASCII art. Unfortunately, her sight is not as it once was, and she has difficulty seeing the pictures.

Moreover, she cannot merely increase the font size as she usually does, because doing so makes the characters stand out more on their own and less as a whole. You come up with a solution: writing a program which duplicates the ASCII lines in order to make them look *bold*.

The first line of the input will contain n, the number of test cases. For each test case, the first line will contain the positive integers w and h, both of which will be at most 100. h lines will follow, each containing w ASCII characters.

The character "." (dot) will denote the background ASCII "color", and the only other color will be "*" (star). For each image, add a 1-character wide border of dots around the image, and proceed to replace every single star character in the original picture with a
```
 *
***
 *
```
, where the middle star is positioned where the original star was.

In the output, separate each test case with "---".

sample input:	**sample output:**
2	.*..
2 1	***.
*.	.*..
3 3	---
.*.	..*..
.	.***.
.*.	*****
	.***.
	..*..

Problem F Cover the Square

Oh no! The Enterprise is being attacked by a Borg unit cube. We all know that resistance is futile, but that won't prevent the captain from trying anyway!

The engineers aboard the Enterprise have recently upgraded the size of the ships's laser beam weapons, such that it can now cut large circular holes into enemy ships. In fact, they can even cut out holes of diameter one, which should get rid of most of the Borg cube in one deadly blow. However, the Borgs are frighteningly persistent, and the captain fears that the remaining undestroyed pieces will cause trouble. Thus, he wants to destroy at least n percent of the cube.

The Borg cube is positioned such that one of its faces is perpendicular to the enterprise's line of sight, so the problem is approximately equivalent to trying to cover a unit square with circles of diameter one. Under this approximation and assuming that the undestroyed parts simply remain motionless at their original position, how many shots are required to destroy at least n percent of the cube?

The first line of the input will contain m, the number of test cases. m lines will follow, each containing an integer n between 0 and 100 inclusively. For each test case, output the number of circles required to cover n percent of the square.

sample input:	**sample output:**
3 0 50 100	0 1 4

Problem G

Bus Schedules

The Association of Commuters in Montreal (ACM) wishes to create a website for the city's public-transit commuters, in order to promote public transit. A prominent reason for people to drive to work instead of commuting is the time wasted on the subway and buses. For this reason, the ACM wishes to add a form on their website so that visitors will be able to specify two points on the island, and the website will find the quickest route between those two points using its database of subway and bus schedules.

Seeing how this may help improve the environment and the greenhouse effect, you offer your help.

The first line of each test case will contain a positive integer n, the number of bus and subway schedules which will follow. Each schedule will begin with a line containing a positive integer m, the number of stops along the path. m lines will follow, describing the stops of the day in chronological order. Each stop will begin by a time in the format $hh : mm$, between $00 : 00$ and $23 : 59$ inclusively. There will be at least one minute between each stop — in other words, all the stop times for a particular bus will be different. A single space will follow, and the rest of the line will contain a name describing the stop.

The name will not contain spaces nor capital letters, and will be at most 20 characters long. Stops with the same name obviously denote the same physical location, where passengers can wait for other buses or subways to stop. After the day completes, the buses and subways mysteriously disappear and reappear at some point before their first stop. They cannot carry any passengers at that time, so the passengers must spend the night waiting at some stop. Each schedule repeats itself every day.

After the schedules, a line will contain a time and two locations of at most 20 characters, the start and the goal. Output the minimum number of minutes needed for a passenger at the start location at the given time to reach the goal location. He is able to enter any bus which stops at his start location at the given starting time or later, and he can also switch from a bus to another instantaneously if they happen to stop at the same place at the same time. He can also wait at a stop for an arbitrary amount of time.

Problem G Bus Schedules

If the destination cannot be reached, output "impossible".

The last line of the input will contain the integer 0 and should not be processed.

All the numbers in the input will be at most 1000.

sample input:	**sample output:**
2 4 00:01 loc_a 00:02 loc_b 00:10 loc_c 00:20 loc_a 2 00:02 loc_b 00:04 loc_c 00:00 loc_a loc_c 1 3 00:00 foo 01:00 bar 02:00 baz 01:30 bar foo 1 4 00:00 baz 01:00 foo 02:00 bar 03:00 baz 02:30 bar foo 0	4 impossible 2790

Problem H Fibonacci Numbers

The Fibonacci sequence is the sequence of numbers such that every element is equal to the sum of the two previous elements, except for the first two elements f_0 and f_1 which are respectively zero and one.

What is the numerical value of the nth Fibonacci number?

For each test case, a line will contain an integer i between 0 and 10^8 inclusively, for which you must compute the ith Fibonacci number f_i. Fibonacci numbers get large pretty quickly, so whenever the answer has more than 8 digits, output only the first and last 4 digits of the answer, separating the two parts with an ellipsis ("...").

Leonardo Fibonacci

There is no special way to denote the end of the of the input, simply stop when the standard input terminates (after the EOF).

sample input:	**sample output:**
0	0
1	1
2	1
3	2
4	3
5	5
35	9227465
36	14930352
37	24157817
38	39088169
39	63245986
40	1023...4155
64	1061...7723
65	1716...7565

Problem H Fibonacci Numbers