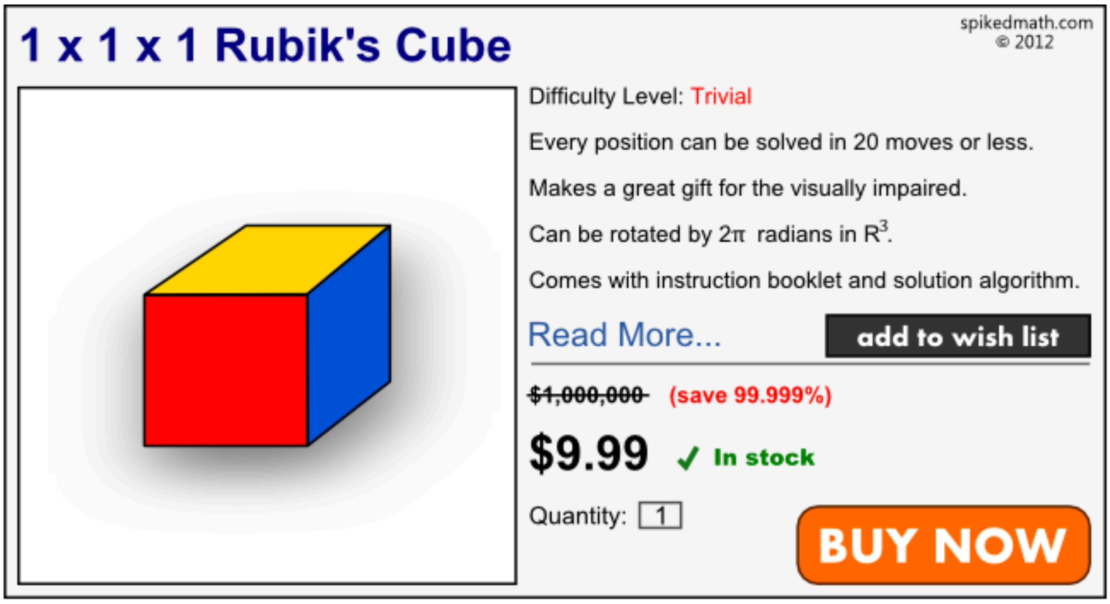
2015 Australian and New Zealand Algorithmic Competition

Round 1
March 21
Warm-up



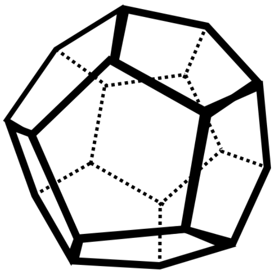
Problems collated (but not authored) by Tim French.

Instructions

- All official entrants must compete in a supervised environment.
- Teams should have between 1 and 3 students, and no access to the internet (beyond the judge server).
- For all problems, read the input from standard input, and write results to standard output.
- The time limit for each problem varies, and is indicated at the start of each question.
- Solutions may be submitted in Python, Java, C or C++. There is no guarantee that all problems will be solvable in the limit using Python, so use it at your own discretion.

Problem A – limit 5 seconds

Polyhedra



Given a sphere, you can slice through the surface of the sphere to make different convex polyhedra. All of these convex polyhedra have the Euler characteristic which can be defined as follows:

$$x = V - E + F = 2$$

where V represents the number of vertices, E the number of edges and F the number of faces on a convex polyhedron.

Input

Input begins with a line with a single integer T, $1 \leq T \leq 100$, denoting the number of test cases. Each test case consists of a single line with two space-separated integers V and E ($4 \leq V, E \leq 100$), representing the number of vertices and edges respectively of the convex polyhedron.

Output

For each test case, print on a single line the number of faces in the defined polyhedron.

Sample Input	Sample Output
2 8 12 4 6	6 4

Problem B – limit 5 seconds

Runes

You are helping an archaeologist decipher some runes. He knows that this ancient society used a Base 10 system, and that they never start a number with a leading zero. He's figured out most of the digits as well as a few operators, but he needs your help to figure out the rest.

The professor will give you a simple math expression. He has converted all of the runes he knows into digits. The only operators he knows are addition (`+`), subtraction (`-`), and multiplication (`*`), so those are the only ones that will appear. Each number will be in the range from $-999,999$ to $999,999$, and will consist of only the digits '`0`'–'`9`', possibly a leading '`-`', and a few '`?`'s. The '`?`'s represent a digit rune that the professor doesn't know (never an operator, an '`=`', or a leading '`-`'). All of the '`?`'s in an expression will represent the same digit (0–9), and it won't be one of the other given digits in the expression.

Given an expression, figure out the value of the rune represented by the question mark. If more than one digit works, give the lowest one. If no digit works, well, that's bad news for the professor—it means that he's got some of his runes wrong. Output -1 in that case.

Input

The sample data will start with the number of test cases T ($1 \le T \le 100$). Each test case will consist of a single line, of the form:

[number][op][number]=[number]

Each [number] will consist of only the digits '`0`'-'`9`', with possibly a single leading minus '`-`', and possibly some '`?`'s. No number will begin with a leading '`0`' unless it is 0, no number will begin with `-0`, and no number will have more than 6 characters (digits or `?`s). The [op] will separate the first and second [number]s, and will be one of: `+`, `-` or `*`. The `=` will always be present between the second and third [number]s. There will be no spaces, tabs, or other characters. There is guaranteed to be at least one `?` in every equation.

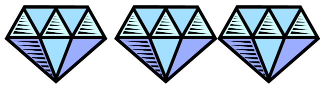
Output

Output the lowest digit that will make the equation work when substituted for the `?`s, or output -1 if no digit will work. Output no extra spaces or blank lines.

Sample Input	Sample Output
`5`	`2`
`1+1=?`	`6`
`123*45?=5?088`	`0`
`-5?*-1=5?`	`-1`
`19--45=5?`	`5`
`??*??=302?`	

Problem C - limit 5 seconds

Diamonds



A diamond's overall worth is determined by its mass in carats as well as its overall clarity. A large diamond with many imperfections is not worth as much as a smaller, flawless diamond. The overall clarity of a diamond can be described on a scale from 0.0–10.0 adopted by the American Gem Society, where 0.0 represents a flawless diamond and 10.0 represents an imperfect diamond.

Given a sequence of N diamonds, each with weight, w_i, in carats and clarity, c_i, on the scale described above, find the longest subsequence of diamonds for which the weight and clarity are both becoming strictly more favorable to a buyer.

Example

In the following sequence of diamonds,

w_i	c_i
1.5	9.0
2.0	2.0
2.5	6.0
3.0	5.0
4.0	2.0
10.0	5.5

the longest desirable subsequence is

1.5	9.0
2.5	6.0
3.0	5.0
4.0	2.0

because the weights strictly increase while the clarities strictly decrease.

Input

Input begins with a line with a single integer T, $1 \leq T \leq 100$, indicating the number of test cases. Each test case begins with a line with a single integer N, $1 \leq N \leq 200$, indicating the number of diamonds. Next follow N lines with 2 real numbers w_i and c_i, $0.0 \leq w_i, c_i \leq 10.0$, indicating the weight in carats and the clarity of diamond i, respectively.

Output

For each test case, output a single line with the length of the longest desirable subsequence of diamonds.

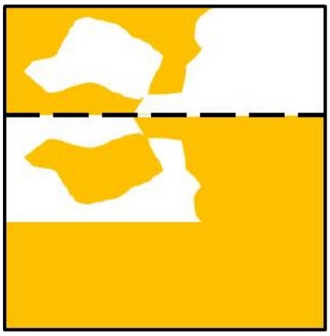
Sample Input	Sample Output

<pre>3
2
1.0 1.0
1.5 0.0
3
1.0 1.0
1.0 1.0
1.0 1.0
6
1.5 9.0
2.0 2.0
2.5 6.0
3.0 5.0
4.0 2.0
10.0 5.5</pre>
	<pre>2
	1
	4</pre>

Problem D – limit 5 seconds

Gold Leaf

Gold leaf is a very thin layer of gold with a paper backing. If the paper gets folded and then unfolded, the gold leaf will stick to itself more readily than it will stick to the paper, so there will be patches of gold and patches of exposed paper. Note that the gold leaf will always stick to itself, rather than the paper. In the following example, the paper was folded along the dashed line. Notice how the gold leaf always sticks to one side or the other, never both.



Consider a crude digital image of a sheet of gold leaf. If the area covered by a pixel is mostly gold, that will be represented by a '#'. If it's mostly exposed paper, it will be represented by a '.'. Determine where the sheet was folded. The sheet was folded exactly once, along a horizontal, vertical, or 45 degree diagonal line. If the fold is horizontal or vertical, it is always between rows/columns. If the fold is diagonal, then the fold goes through a diagonal line of cells, and the cells along the fold are always '#'.

Input

Input will start with a single line containing the number of cases, between 1 and 100, inclusive. Each test case will begin with a line with two integers, N and M $2 \leq N, M \leq 25$, where N is the number of rows, and M is the number of columns of the photograph. Each of the next N lines will contain exactly M characters, all of which will be either '#' or '.'. This represents a crudely represented digital image of the sheet of gold leaf. There is guaranteed to be at least one '.', and there is guaranteed to be a solution.

Output

For each test case, output four integers, indicating the places where the fold hits the edges of the paper. Output them in the order r1 c1 r2 c2 where (r1,c1) and (r2,c2) are row/column coordinates (r=row, c=column). The top left character is (1,1) and the bottom right is (n,m). If the fold is horizontal or diagonal, list the left side coordinates before the right. If the fold is vertical, list the top coordinates before the bottom. If the fold is horizontal, use the coordinates above the fold. If the fold is vertical, use the coordinates to the left of the fold. If the fold is diagonal, use the coordinates of the cells that the fold goes through. If more than one fold is possible, choose the one with the smallest first coordinate, then the smallest second coordinate, then third, then fourth.

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>
<pre>
3
8 10
#.#..##..#
####..####
###.##....
...#..####
....##....
.#.##..##.
##########
##########
5 20
###########.#.#.#.#.
###########...#.###.
##########..##.#..##
###########..#.#.##.
###########.###...#.
5 5
.####
###.#
##..#
#..##
#####
</pre>
</td><td>
<pre>
3 1 3 10
1 15 5 15
4 1 1 4
</pre>
</td></tr>
</table>

Problem E – limit 5 seconds

Ramp Number

A Ramp Number is a number whose digits only rise or stay the same; they never fall.

- 123 is a ramp number.
- 101 is not a ramp number.
- 1111000001111 is not a ramp number.

Given a positive integer n, if it is a ramp number, print the number of ramp numbers less than it. If it is not a ramp number, print `-1`.

Input

Input will start with a single line giving the number of test cases. Each test case will be a single positive integer on a single line, with up to 80 digits. The result will always fit into a 64-bit long.

Output

For each test case, print `-1` if the input is not a ramp number. Print the number of ramp numbers less than the input value if the input value is a ramp number.

Sample Input	Sample Output
5 11 123 101 1111 99999	10 65 -1 220 2001

Problem F – limit 5 seconds

Wrench

Peter works at a factory. He is looking at a list of wrench sizes and needs to find the appropriately sized wrench for various screws and nuts and bolts to do his work. Normally, these sizes are specified using US Customary Unit notation such as 13/16", or 3/8", and so on.

Another way to write 13/16" is 0.8125"

But the reference sheets for various parts round the numbers in weird ways, and give approximations only, so for example 13/16" might turn into 0.812, or 0.813, or sometimes just 0.81, depending on the method of rounding.

Given that Peter is looking for a wrench of size A/B", and it is customary for B to be a power of 2, help Peter find the correct wrench size, where A is a positive integer and B is the minimum possible base (power of 2).

Input

Input starts with the number of test cases, T, on a single line, with $1 \leq T \leq 100$. Each test case is a single decimal number on its own line representing a wrench size, with at most six digits after the decimal point. There need not always be a decimal point. The input value will be greater than zero.

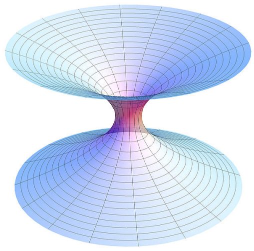
Output

A/B", or C A/B", or C", where B is the minimal power of two such that the exact decimal representation rounded to the number of decimal digits of the input matches the input, using one of the following rounding rules: round up (ceiling), round down (or truncate), or round-to-nearest. The wrench will be less than or equal to 10 inches. There will always be a valid power of two less than or equal to 128.

Sample Input	Sample Output
10 0.81 .8125 0.37 2 2.4 2.99 2.40 1.27 4. 9.242187	13/16" 13/16" 3/8" 2" 2 3/8" 2 63/64" 2 13/32" 1 17/64" 4" 9 31/128"

Problem G – limit 5 seconds

Wormhole



With our time on Earth coming to an end, Cooper and Amelia have volunteered to undertake what could be the most important mission in human history: travelling beyond this galaxy to discover whether mankind has a future among the stars. Fortunately, astronomers have identified several potentially inhabitable planets and have also discovered that some of these planets have wormholes joining them, which effectively makes the travel distance between these wormhole connected planets zero. For all other planets, the travel distance between them is simply the Euclidean distance between the planets. Given the location of Earth, planets, and wormholes, find the shortest travel distance between any pairs of planets.

Input

- The first line of input is a single integer, T ($1 \leq T \leq 10$) the number of test cases.
- Each test case consists of planets, wormholes, and a set of distance queries.
- The planets list for a test case starts with a single integer, p ($1 \leq p \leq 60$), the number of planets. Following this are p lines, where each line contains a planet name along with the planet's integer coordinates, i.e. *name x y z* ($0 \leq x, y, x \leq 2 \cdot 10^6$) The names of the planets will consist only of ASCII letters and numbers, and will always start with an ASCII letter. Planet names are case-sensitive (Earth and earth are distinct planets). The length of a planet name will never be greater than 50 characters. All coordinates are given in parsecs.
- The wormholes list for a test case starts with a single integer, w ($0 \leq w \leq 40$), the number of wormholes, followed by the list of w wormholes. Each wormhole consists of two planet names separated by a space. The first planet name marks the entrance of wormhole, and the second planet name marks the exit from the wormhole. The planets that mark wormholes will be chosen from the list of planets given in the preceding section. Note: you can't enter a wormhole at its exit.
- The queries list for a test case starts with a single integer, q ($1 \leq q \leq 20$), the number of queries. Each query consists of two planet names separated by a space. Both planets will have been listed in the planet list.

Output

For each test case, output a line, "`Case` i`:`", the number of the ith test case. Then, for each query in that test case, output a line that states "`The distance from planet`$_1$ `to planet`$_2$ `is` d `parsecs.`", where the planets are the names from the query and d is the shortest possible travel distance between the two planets. Round d to the nearest integer.

Sample Input	Sample Output
3 4 Earth 0 0 0 Proxima 5 0 0 Barnards 5 5 0 Sirius 0 5 0 2 Earth Barnards Barnards Sirius 6 Earth Proxima Earth Barnards Earth Sirius Proxima Earth Barnards Earth Sirius Earth 3 z1 0 0 0 z2 10 10 10 z3 10 0 0 1 z1 z2 3 z2 z1 z1 z2 z1 z3 2 Mars 12345 98765 87654 Jupiter 45678 65432 11111 0 1 Mars Jupiter	Case 1: The distance from Earth to Proxima is 5 parsecs. The distance from Earth to Barnards is 0 parsecs. The distance from Earth to Sirius is 0 parsecs. The distance from Proxima to Earth is 5 parsecs. The distance from Barnards to Earth is 5 parsecs. The distance from Sirius to Earth is 5 parsecs. Case 2: The distance from z2 to z1 is 17 parsecs. The distance from z1 to z2 is 0 parsecs. The distance from z1 to z3 is 10 parsecs. Case 3: The distance from Mars to Jupiter is 89894 parsecs.

Problem H – limit 60 seconds

Knights

Magnus is the youngest chess grandmaster ever. He loves chess so much that he has decided to decorate his home with chess pieces. To decorate his long corridor, he has decided to use knights. His corridor is covered by beautiful marble squares of M rows and N columns. He wants to put the knights pieces into some (or possibly none) of these cells. Each cell will contain at most one knight.

The special thing in his arrangement is no pair of knights can attack each other. (Two knights can attack each other if they are placed in opposite corner cells of a 2 by 3 rectangle).

Given the dimension of the long corridor, your task is to calculate how many ways Magnus can arrange his knight pieces. Two arrangements are considered different if there exists a cell which contains a knight in the first arrangement but not in the other arrangement.

Input

The first line of the input is the number of test cases T ($T \leq 10$). Then T test cases follow. Each test case consists of 2 numbers: M, the number of rows, and N, the number of columns. ($1 \leq M \leq 4, 1 \leq N \leq 10^9$).

Output

For each test case, print the number of possible ways modulo 1,000,000,009 ($10^9 + 9$)

Sample Input	Sample Output
4 1 2 2 2 3 2 4 31415926	4 16 36 413011760

Problem I – limit 5 seconds

Number Game

Alice and Bob are playing a game on a line of N squares. The line is initially populated with one of each of the numbers from 1 to N. Alice and Bob take turns removing a single number from the line, subject to the restriction that a number may only be removed if it is not bordered by a higher number on either side. When the number is removed, the square that contained it is now empty. The winner is the player who removes the 1 from the line. Given an initial configuration, who will win, assuming Alice goes first and both of them play optimally?

Input

Input begins with a line with a single integer T, $1 \leq T \leq 100$, denoting the number of test cases. Each test case begins with a line with a single integer N, $1 \leq N \leq 100$, denoting the size of the line. Next is a line with the numbers from 1 to N, space separated, giving the numbers in line order from left to right.

Output

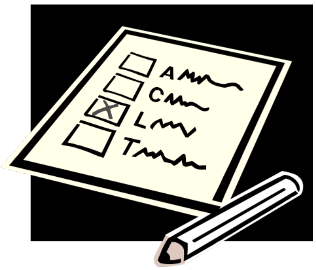
For each test case, print the name of the winning player on a single line.

Sample Input	Sample Output

`4`
`4`
`2 1 3 4`
`4`
`1 3 2 4`
`3`
`1 3 2`
`6`
`2 5 1 6 4 3`
	`Bob`
	`Alice`
	`Bob`
	`Alice`

Problem J – limit 5 seconds

Majority



The votes are in! Mathematicians world-wide have been polled, and each has chosen their favorite number between 1 and 1000. Your goal is to tally the votes and determine what the most popular number is.

If there is a tie for the greatest number of votes, choose the smallest number with that many votes.

Input

Input will start with a single line containing the number of test cases, between 1 and 100 inclusive. For each test case, there will be a single line giving the number of votes V, $1 \leq V \leq 1000$. Following that line will be V lines, each with a single integer vote between 1 and 1000.

Output

Sample Input	Sample Output
3 3 42 42 19 4 7 99 99 7 5 11 12 13 14 15	42 7 11