Pacific Northwest Region Programming Contest Division 1

November 15th, 2014

Reminders

- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Platform is as follows:

  ```
  Ubuntu 14.04.1 LTS x86_64
  geany
  java version 1.7.0_65
  c/c++ gcc version 4.8.2
  eclipse 4.4 with CDT 8.4
  Python 2.7.6 (IDE support)
  Python 3.4.0 (syntax highlighting editor support)
  ```

- Compiler options are as follows:

  ```
   g++ -g -O2 -std=gnu++0x -static $*
   gcc -g -O2 -std=gnu99 -static $* -lm
   javac -encoding UTF-8 -sourcepath . -d . $* runjava
   java -client -Xss8m -Xmx1024m $*
   python $*
  ```

- Python may not have sufficient performance for many of the problems; use it at your discretion.

Problem A — limit 5 seconds

Runes

ᚱᚢᚾᛟᚷᛖᚠ

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 ($1 \leq T \leq 100$). 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

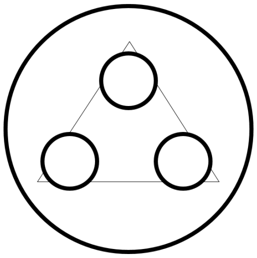
Output the lowest digit that will make the equation work when substituted for the `?`s, or output -1 if no digit will work. Output no extra spaces or blank lines.

Sample Input	**Sample Output**

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

Problem B — limit 20 seconds

Alchemy



Since the days of yore alchemy has been studied and practiced. The practice makes alchemists able to transmute materials into other forms. Transmuting materials requires drawing a transmutation circle on the ground. A little known fact about transmutation circles is they can be drawn inside other transmutation circles. By activating certain configurations in the correct order, more powerful transmutations can be produced. Activating circles incorrectly can have drastic effects on the alchemist's body.

A young alchemist named Nicholas Flamel would like to learn the ways of alchemy. He has drawn several configurations of transmutation circles on the ground. When a circle is activated it burns bright red representing the element of fire. The activation itself produces no extra energy. The secret is when an outer transmutation circle is activated. When this occurs, all already active circles in the area of the activated circle quickly change to their respective complement elements. Fire changes to a cool blue representing water. Circles that were blue for water will burn fiery red once again. This change can either create or drain energy from the transmutation. Beware, energy can go negative at any time by temporarily draining the alchemist's life force (but the spell continues to work just fine).

Nicholas wants to get as much out of his transmutations as possible. To do so requires him to activate all his circles in an order that releases the most energy. Determine the maximum amount of energy that can be released.

Input

Input will start with a single line with an integer T giving the number of test cases, between 1 and 100 inclusive. For each set of transmutation circles there will be a line with a number N, $1 \leq N \leq 2000$. This represents the number of transmutation circles. The next N lines contain spaced-separated integers X Y R A B. The first three integers represent the coordinates and radius of the circle, while the last two represent the energy release for going from fire to water (A), and water to fire (B). $-10{,}000 \leq X, Y \leq 10{,}000$; $1 \leq R \leq 10{,}000$; $-500 \leq A, B \leq 500$.

No two circles will intersect or touch.

Output

For each set of transmutation circles print a single integer representing the maximum energy that can be produced by activating the circles. On the following line print the permutation of the input circles that can produce that energy. If multiple permutations exist, print the one that comes first lexicographically.

Sample Input

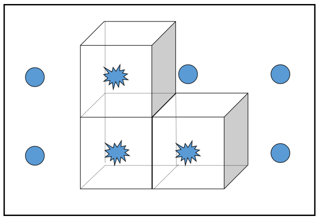
```
1
8
0 0 100 -100 -100
0 0 50 -10 -10
0 0 10 -100 500
0 0 1 100 100
1000 1000 100 -1 1
1000 1000 50 -1 1
1000 1000 10 -1 1
1000 1000 1 -1 1
```

Sample Output

```
700
4 3 1 2 5 6 7 8
```

Problem C — limit 30 seconds

Containment



A three-dimensional grid of atomic energy cells aboard the starship PacificNorthwestPassage is reporting failures on several of its cells. The ship's engineer must set up enclosures that will contain all cells reported to be failing in order to avoid a meltdown. It is imperative that the enclosures be finished in the shortest amount of time, even if that requires some healthy cells to be enclosed along with the defective ones. The enclosures are formed by panels which can only be inserted between cells (so each individual panel must be axis-aligned), and each panel separates exactly two cells and requires one minute to set up. For full containment, each enclosure must form a closed polytope. Given the coordinates of each defective cell, report how long it will take to finish containing the problem.

Input

The input will start with a single line containing the number N giving the number of test cases, $1 \leq N \leq 100$. Each test case will start with a single line giving the number F, $1 \leq F \leq 100$, the number of failing cells. Following this will be F lines each with three integers x_i, y_i, z_i, all between 0 and 9, inclusive, giving the locations of the failing cells. For each test case, each triple (x_i, y_i, z_i) will be unique.

Output

For each test case, print on a single line the minimum number of minutes required to fully contain the problem.

Sample Input

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

Sample Output

```
6
10
14
```

Problem D — limit 5 seconds

Function

Given a function defined by:

$$\text{Let } t = n/m; \qquad z[t] = \begin{cases} (a + (b + rt^2)z[-1 - 2t])/r & \text{if } t \leq 0 \\ (c + (d + rt^2)z[1 - 2t])/r & \text{if } t > 0 \end{cases}$$

Given integer a, b, c, d, n, m, and r, evaluate $z[t]$ (as a floating point number).

$$\begin{gathered} 1 \leq n \leq m \leq 100 \\ 1 \leq b \leq r \\ 1 \leq d \leq r \\ 1 \leq r \leq 1000 \\ 1 \leq a \leq 1000 \\ 1 \leq c \leq 1000 \end{gathered}$$

There will be a solution. Your result must be accurate to within $\pm 10^{-6}$ absolute error, or $\pm 10^{-6}$ relative error.

Input

On the first line will be the number of functions to solve, between 1 and 100 inclusive. Following that will be one line per function, giving the integer parameters separated by spaces in the order n, m, a, b, c, d, r.

Output

The value of the $z[n/m]$ should be printed in floating point format, one line per parameter set.

Sample Input	**Sample Output**

`3`
`1 1 1 1 1 1 1`
`2 3 1 2 3 4 10`
`2 3 5 6 7 8 9`
	`-1`
	`0.4225806452`
	`4.111111111`

Problem E — limit 5 seconds

Hill Number

A Hill Number is a number whose digits possibly rise and then possibly fall, but never fall and then rise.

- 12321 is a hill number.
- 101 is not a hill number.
- 1111000001111 is not a hill number.

Given an integer n, if it is a hill number, print the number of hill numbers less than it. If it is not a hill 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 70 digits. The result will always fit into a 64-bit long.

Output

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

Sample Input

```
5
10
55
101
1000
1234321
```

Sample Output

```
10
55
-1
715
94708
```

Problem F — 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 G — 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

```
4
4
2 1 3 4
4
1 3 2 4
3
1 3 2
6
2 5 1 6 4 3
```

Sample Output

```
Bob
Alice
Bob
Alice
```

Problem H — limit 5 seconds

Pushups

A friend of yours is on the cheer squad for their football team. Each time the team scores, the cheer squad does pushups—one for each point the team has scored so far. If the teams scores a touchdown (7 points), the squad does 7 pushups. If the team then scores a field goal (3 points), the cheer squad does 10 pushups. If the team then scores a safety (2 points), the squad will do 12 pushups. At the end of that game, the squad will end up having done 7+10+12=29 pushups!

You meet your friend after a game, and they say "Boy, am I tired! I did a total of n pushups at the game today!" and promptly collapse from exhaustion. Given n, the number of pushups, can you figure out how the team scored? More than one score may be possible. For example, for 29 pushups, the team could have scored 3, then 2, then 2, then 7, for a total of 14 points. If so, find the highest possible score.

Input

The input will start with a single number on the first line giving the number of test cases, between 1 and 20, inclusive. Each test case will begin with two integers N and M $1 \leq N \leq 5{,}000, 1 \leq M \leq 10$ where N is the number of pushups the cheer squad did, and M is the number of ways a team can score points in that sport. On the next line will be M unique integers S_i, $1 \leq S_i \leq 20$, with a single space between them, indicating the number of points the team gets for each kind of score. The scores are independent; a team can accrue scores in any order.

Output

For each test case, output a single integer indicating the team's final score. If more than one final score can lead to the given number of pushups, output the largest one. If no final score can lead to the given number of pushups, then your friend must have miscounted. In this case, output '-1'. Output no extra spaces.

Sample Input	**Sample Output**

4
29 3
7 3 2
15 1
1
16 1
1
6 2
3 1
	14
	5
	-1
	3

Problem I — limit 10 seconds

Salary Inequity

There is a large company of N employees. With the exception of one employee, everyone has a direct supervisor. The employee with no direct supervisor is indirectly a supervisor of all other employees in the company. We say that employee X is a subordinate of employee Y if either Y is the direct supervisor of X, or the direct supervisor of X is a subordinate of Y.

One day, the HR department decides that it wants to investigate how much inequity there is in the company with respect to salaries. For a given employee, the inequity of the employee is the difference between the minimum salary of that employee and all his/her subordinates and the maximum salary of that employee and all his/her subordinates.

HR wants to be able to compute the inequity for any employee quickly. However, this is complicated by the fact that an employee will sometimes give himself/herself, along with all his/her subordinates, a raise. Can you help?

Input

The first line of your input file contains a number T representing the number of companies you will be analyzing for inequity. T will be at most 20.

For each company, there will be a line containing an integer N, representing the number of employees at the company. Each employee is assigned an ID which is a unique integer from 1 to N. The next line will contain $N - 1$ integers. The Kth integer in that line is the ID of the direct supervisor of employee $(K + 1)$. The next line will contain N integers, the Kth integer in this line being the salary of employee K. The next line contains an integer Q, the number of events that you will need to process. There are two types of events to process - raises and inequity queries. In the event of a raise, the line will start with the letter R followed by the ID of the employee and an integer representing the increase in salary for that employee and all his/her subordinates. In the event of an inequity query, the line will start with the letter Q followed by the ID of the employee for whom inequity needs to be determined.

$2 <= N <= 1{,}000{,}000$

$1 <= Q <= 10{,}000$

For every employee K, the ID of his/her supervisor will be less than K. Initial salaries will range from 1 to 1,000. No raise will exceed 1,000.

Output

For each inequity query, print the inequity of the employee on its own line.

Sample Input

```
1
5
1 1 2 2
10 6 8 4 5
7
Q 2
Q 3
R 4 2
Q 2
Q 1
R 2 4
Q 1
```

Sample Output

```
2
0
1
5
2
```

Problem J — limit 10 seconds

Stamp Stamp

Bureaucrats love bureaucracy. This assertion seems fairly obvious but a less obvious observation is the amount of paperwork this means! When paperwork is complete, the bureaucrat stamps the official document with their official stamp of office. Some bureaucrats are extra thorough and stamp the document multiple times.

We are interested primarily in bureaucrats that stamp their documents twice. A bureaucrat stamp takes up some rectangular area.

For example, below is a bureaucrat's stamp:

```
..#..#..
.######.
..#..#..
```

When the bureaucrat stamps the paper twice, it is potentially moved to different location on the paper but not rotated. The stamp will always be axis aligned.

The '#' symbol on a stamp covers the paper with black ink at the cell on the paper that is pressed. A '.' doesn't leave any marks on the paper nor does it remove a mark. If a cell of the paper is marked twice it is undiscernable from a cell that is marked once.

You will be given a paper that was stamped twice by a stamp. Your task is to determine the minimum number of nubs (# symbols) that could have possibly been on the original stamp. The paper is guaranteed to be stamped twice by the entire stamp. (All of the stamp will be on the paper in both stampings)

For example the above stamp could have made the following mark on paper:

```
..#..#..
.######.
.######.
..#..#..
```

Input

You are given a number T representing the number of pieces of paper to analyze, $(1 \leq T \leq 100)$. For each piece of paper, there will be two numbers L and W representing the length and width of the paper, $(1 <= W, L <= 300)$. The next L lines will each contain a string of length W. This string will consist of only the characters '.' and '#'. '.' represents a white cell of paper. '#' represents a cell covered in black ink.

At most 15 of the test cases will have a width or length greater than 80.

Output

For each piece of paper, output the minimum number of nubs possible on the original stamp.

Sample Input

```
5
4 8
..#..#..
.######.
.######.
..#..#..
3 3
...
.#.
...
2 6
.#####
#####.
2 5
.#.#.
#.#.#
6 6
###.##
#.####
######
######
#.####
######
```

Sample Output

```
8
1
5
3
21
```

Problem K — limit 10 seconds

Towers

The Towers puzzle challenges a single player to place towers of varying heights in an n x n grid ($3 \leq n \leq 5$). The heights of each tower can be any integer between 1 and n, inclusive, but the placement of the n^2 towers must be such that no tower of the same height appears twice within the same row or column. Given no other constraints, there is an exponentially large number of ways to place towers. Take, for example, the 5 x 5 puzzle, where one of the solutions looks like this:

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

The puzzles become more interesting (and harder to solve) as they further constrain that one or more grid locations be occupied by towers of specified heights. A 5 x 5 puzzle, for instance, might require that the upper left and lower right corners house towers of height 3, and that the center location house a tower of height 5. The puzzle would look like that on the left below, and a solution—again, one of many—might look like that presented to its right.

3				
		5		
				3

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

Some puzzles include one or more numbers around the perimeter, where each number specifies the exact number of towers visible when looking into the grid from that direction, with the understanding that taller towers fully conceal shorter ones. For example, the 5 x 5 puzzle presented below and on the left would have the (incidentally unique) solution to its right.

					1	
2			3			3
2						2
	1					
				1		2

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

The number above the final column requires that just a single tower be visible when viewing its sequence of five towers from above (which essentially means that the top row of that column must house a 5.) The second row introduces multiple requirements:

- the center column must house a 3,
- exactly two towers are visible when viewing its tower sequence from the left, and
- exactly three towers are visible when viewing its two sequence from the right.

Input

The input starts with a single integer on a line by itself, giving the number of tests; there will be at least 1 but no more than 100 test cases. Each $n \times n$ puzzle ($3 \leq n \leq 5$) is expressed as a series of $n + 2$ lines of length $n + 2$. The outer perimeter of the grid specifies the visibility constraints (where '`-`' expresses there are no constraints for that row or column from the relevant direction looking in; the corners of the perimeter are always '`-`'] and the interior of the grid specifies those locations where a tower of a specific height must be placed (where '`-`' expresses there is no imposed tower height for that location.)

We guarantee that every character in the grid is either a '`-`' or a digit between 1 and n.

Output

For each input puzzle, output a solution as a sequence of n lines, each of length n, followed by a blank line. If a puzzle has multiple solutions, then output any one of them. If a puzzle cannot be solved, simply print the word "`no`", all by itself, without the delimiting double quotes, followed by a blank line.

Sample Input

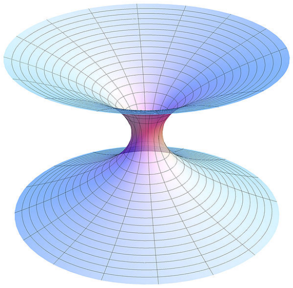
```
5
5
-------
-------
-------
-------
-------
-------
-------
5
-41223-
2-----3
3-----2
2-----1
1-----5
3-----2
-23212-
3
-111-
----3
2---2
----2
-131-
5
--33---
------3
-------
------3
-------
3------
--324--
5
-----1-
-------
2--3--3
2-----2
-1-----
----1-2
-------
```

Sample Output

```
12345
21453
34512
45231
53124

15243
23514
42135
54321
31452

no

51243
24351
45132
13524
32415

31425
25341
42153
14532
53214
```

Problem L — 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

```
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
```

Sample Output

```
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.
```