Problem A. Read Phone Number

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 6 points	Solve A-small
Large input 13 points	Solve A-large

Problem

Do you know how to read the phone numbers in English? Now let me tell you.

For example, In China, the phone numbers are 11 digits, like: 15012233444. Someone divides the numbers into 3-4-4 format, i.e. 150 1223 3444. While someone divides the numbers into 3-3-5 format, i.e. 150 122 33444. Different formats lead to different ways to read these numbers:

150 1223 3444 reads one five zero one double two three three triple four.

150 122 33444 reads one five zero one double two double three triple four.

Here comes the problem:

Given a list of phone numbers and the dividing formats, output the right ways to read these numbers.

Rules:

Single numbers just read them separately.

2 successive numbers use double.

3 successive numbers use triple.

4 successive numbers use quadruple.

5 successive numbers use quintuple.

6 successive numbers use sextuple.

7 successive numbers use septuple.

8 successive numbers use octuple.

9 successive numbers use nonuple.

10 successive numbers use decuple.

More than 10 successive numbers read them all separately.

Input

The first line of the input gives the number of test cases, **T**. **T** lines|test cases follow. Each line contains a phone number **N** and the dividing format **F**, one or more positive integers separated by dashes (-), without leading zeros and whose sum always equals the number of digits in the phone number.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the reading sentence in English whose words are separated by a space.

Limits

1 ≤ **T** ≤ 100.

Small dataset

1 ≤ **length of N** ≤ 10.

Large dataset

1 ≤ **length of N** ≤ 100.

Sample

Input

```
3
15012233444 3-4-4
15012233444 3-3-5
```

Submissions

Read Phone Number

6pt	Not attempted **1885/3058 users** correct (62%)
13pt	Not attempted **1094/1837 users** correct (60%)

Rational Number Tree

9pt	Not attempted **1193/1545 users** correct (77%)
12pt	Not attempted **368/1037 users** correct (35%)

Sorting

5pt	Not attempted **1666/1990 users** correct (84%)
8pt	Not attempted **1551/1635 users** correct (95%)

Cross the maze

10pt	Not attempted **134/370 users** correct (36%)
13pt	Not attempted **119/132 users** correct (90%)

Spaceship Defence

10pt	Not attempted **175/382 users** correct (46%)
14pt	Not attempted **106/152 users** correct (70%)

Top Scores

dreamoon	100
springegg	100
tckwok	100
cgy4ever	100
OR.Director	100
AlanC	100
Mochavic	100
jxwuyi	100
oldherl	100
Descent	100

```
12223 2-3
```

Output

```
Case #1: one five zero one double two three three triple fo
Case #2: one five zero one double two double three triple t
Case #3: one two double two three
```

Round A China New Grad Test 2014

A. Read Phone Number
B. Rational Number Tree
C. Sorting
D. Cross the maze
E. Spaceship Defence

Questions asked

Problem B. Rational Number Tree

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 9 points	Solve B-small
Large input 12 points	Solve B-large

Problem

Consider an infinite complete binary tree where the root node is 1/1 and left and right childs of node p/q are p/(p+q) and (p+q)/q, respectively. This tree looks like:

```
         1/1
    ______|______
    |           |
   1/2         2/1
 ___|___     ___|___
 |     |     |     |
1/3   3/2   2/3   3/1
...
```

It is known that every positive rational number appears exactly once in this tree. A level-order traversal of the tree results in the following array:

```
1/1, 1/2, 2/1, 1/3, 3/2, 2/3, 3/1, ...
```

Please solve the following two questions:

1. Find the **n**-th element of the array, where **n** starts from 1. For example, for the input 2, the correct output is 1/2.
2. Given **p/q**, find its position in the array. As an example, the input 1/2 results in the output 2.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of one line. The line contains a problem id (1 or 2) and one or two additional integers:

1. If the problem id is 1, then only one integer **n** is given, and you are expected to find the **n**-th element of the array.
2. If the problem id is 2, then two integers **p** and **q** are given, and you are expected to find the position of **p/q** in the array.

Output

For each test case:

1. If the problem id is 1, then output one line containing "`Case #x: p q`", where x is the case number (starting from 1), and p, q are numerator and denominator of the asked array element, respectively.
2. If the problem id is 2, then output one line containing "`Case #x: n`", where x is the case number (starting from 1), and n is the position of the given number.

Limits

$1 \le \mathbf{T} \le 100$; **p** and **q** are relatively prime.

Small dataset

$1 \le \mathbf{n}, \mathbf{p}, \mathbf{q} \le 2^{16}-1$; **p/q** is an element in a tree with level number ≤ 16.

Large dataset

$1 \le \mathbf{n}, \mathbf{p}, \mathbf{q} \le 2^{64}-1$; **p/q** is an element in a tree with level number ≤ 64.

Sample

Input	Output
4 1 2	Case #1: 1 2 Case #2: 2

```
2 1 2    Case #3: 3 2
1 5      Case #4: 5
2 3 2
```

Round A China New Grad Test 2014

A. Read Phone Number
B. Rational Number Tree
C. Sorting
D. Cross the maze
E. Spaceship Defence

Questions asked

dreamoon	100
springegg	100
tckwok	100
cgy4ever	100
OR.Director	100
AlanC	100
Mochavic	100
jxwuyi	100
oldherl	100
Descent	100

Problem C. Sorting

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 5 points	Solve C-small
Large input 8 points	Solve C-large

Problem

Alex and Bob are brothers and they both enjoy reading very much. They have widely different tastes on books so they keep their own books separately. However, their father thinks it is good to promote exchanges if they can put their books together. Thus he has bought an one-row bookshelf for them today and put all his sons' books on it in random order. He labeled each position of the bookshelf the owner of the corresponding book ('Alex' or 'Bob').

Unfortunately, Alex and Bob went outside and didn't know what their father did. When they were back, they came to realize the problem: they usually arranged their books in their own orders, but the books seem to be in a great mess on the bookshelf now. They have to sort them right now!!

Each book has its own *worth*, which is represented by an integer. Books with odd values of worth belong to Alex and the books with even values of worth belong to Bob. Alex has a habit of sorting his books from the left to the right in an increasing order of worths, while Bob prefers to sort his books from the left to the right in a decreasing order of worths.

At the same time, they do not want to change the positions of the labels, so that after they have finished sorting the books according their rules, each book's owner's name should match with the label in its position.

Here comes the problem. A sequence of $\mathbf{N}$ values $\mathbf{s_0}$, $\mathbf{s_1}$, ..., $\mathbf{s_{N-1}}$ is given, which indicates the worths of the books from the left to the right on the bookshelf currently. Please help the brothers to find out the sequence of worths after sorting such that it satisfies the above description.

Input

The first line of input contains a single integer $\mathbf{T}$, the number of test cases. Each test case starts with a line containing an integer $\mathbf{N}$, the number of books on the bookshelf. The next line contains $\mathbf{N}$ integers separated by spaces, representing $\mathbf{s_0}$, $\mathbf{s_1}$, ..., $\mathbf{s_{N-1}}$, which are the worths of the books.

Output

For each test case, output one line containing "Case #**X**: ", followed by $\mathbf{t_0}$, $\mathbf{t_1}$, ..., $\mathbf{t_{N-1}}$ in order, and separated by spaces. **X** is the test case number (starting from 1) and $\mathbf{t_0}$, $\mathbf{t_1}$, ..., $\mathbf{t_{N-1}}$ forms the resulting sequence of worths of the books from the left to the right.

Limits

$1 \le \mathbf{T} \le 30$.

Small dataset

$1 \le \mathbf{N} \le 100$
$-100 \le \mathbf{s_i} \le 100$

Large dataset

$1 \le \mathbf{N} \le 1000$
$-1000 \le \mathbf{s_i} \le 1000$

Sample

Input	Output

<pre>2
5
5 2 4 3 1
7
-5 -12 87 2 88 20 11</pre>
	<pre>Case #1: 1 4 2 3 5
	Case #2: -5 88 11 20 2 -12 87</pre>

A. Read Phone Number
B. Rational Number Tree
C. Sorting
D. Cross the maze
E. Spaceship Defence

Problem D. Cross the maze

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 10 points	Solve D-small
Large input 13 points	Solve D-large

Problem

Edison, a robot, does not have a right hand or eyes. As a brave robot, **he always puts his left hand on the wall no matter he walks or turns around**. Because he thinks it is too dangerous, Edison does not walk backward.

Assume that Edison has found himself in a square-shaped maze of **N**x**N** square cells which is surrounded by walls from the outside. In the maze, some of the cells are also walls. Edison can only move between two empty cells in four directions, north, south, west and east. In order to get out of the maze, he drafts a plan. He uses his left hand to lean on the wall and goes by following the wall.

Here is the question, is Edison able to get out of the maze in at most 10,000 steps? If he can make it, output the path. By getting out of the maze, he only needs to be in the exit cell. If the starting cell is the same as the exit, Edison won't need to move and can directly get out of the maze.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with an integer **N**. **N** is the size of the maze. The following **N** lines, each line contains **N** characters which may be '.' or '#'. '.' is an empty cell, '#' is a wall. Followed by a line which contains four integers: **sx**, **sy**, **ex**, **ey**. (**sx**, **sy**) means that Edison is standing on row **sx** and column **sy** as his starting cell, (**ex**, **ey**) is the exit of the maze. (**sx**, **sy**) is guaranteed to be at one of the 4 corners of the maze, and Edison can only touch the wall on 4 adjacent cells(not 8) initially. (**ex**, **ey**) can be anywhere in the maze. Note that the top-left corner is at position (1,1).

Output

For each test case, output a line containing "Case #x: y", where x is the case number (starting from 1) and y is "Edison ran out of energy." (without the quotes) if Edison can't reach the exit of the maze in at most 10,000 steps, otherwise y should be the number of steps followed by another line which contains y characters to describe the path (each character should be E for east, S for south, W for west or N for north). There is no character to represent the turning around. We don't care about the turning around steps, please only output the path of how Edison will cross the maze.

Limits

1 ≤ **T** ≤ 30.
1 ≤ **sx**, **sy**, **ex**, **ey** ≤ **N**.
The starting cell and the exit of the maze will always be an empty cell. And the starting cell and the exit of the maze won't be the same.

Small dataset

2 ≤ **N** ≤ 10.

Large dataset

2 ≤ **N** ≤ 100.

Sample

Input	Output

<pre>3
2
.#
#.
1 1 2 2
5
.##.#
.....
...#.
.###.
...#.</pre>
	<pre>Case #1: Edison ran out of energy.
	Case #2: 22
	SEEENSESSSNNNWWSWWSSEE
	Case #3: 4
	EESS</pre>

```
1 1 5 3
3
...
.#.
...
1 1 3 3
```

Note:
In the 2nd test case after moving 1 cell down from his starting cell, Edison will still be able to lean on the wall at the cell (1,2) by his left hand.
In the third test case, due to Edison can't touch the wall at cell (2,2) initially, so he has to go east in his first step.

Round A China New Grad Test 2014

A. Read Phone Number

B. Rational Number Tree

C. Sorting

D. Cross the maze

E. Spaceship Defence

Questions asked

Problem E. Spaceship Defence

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 10 points	Solve E-small
Large input 14 points	Solve E-large

Problem

The enemy has invaded your spaceship, and only superior tactics will allow you to defend it! To travel around your spaceship, your soldiers will use two devices: *teleporters* and *turbolifts*.

Teleporters allow your soldiers to move instantly between rooms. Every room contains a teleporter, and rooms are color-coded: if a soldier is in a room with some color, she can use the teleporter in that room to immediately move to any other room with the same color.

Turbolifts allow your soldiers to move between rooms more slowly. A turbolift is like an elevator that moves in many directions. Each turbolift moves from one room to one other room, and it takes a certain amount of time to travel. Notes about turbolifts:

- Turbolifts are not two-way: if a turbolift moves soldiers from room a to room b, the same turbolift cannot move soldiers from room b to room a, although there might be another turbolift that does that.
- More than one soldier can use the same turbolift, and they do not interfere with each other in any way.

You will be given the locations and destinations of several soldiers. For each soldier, output the minimum amount of time it could take that soldier to travel from his location to his destination.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow.

For every test case:

The first line of every test case contains an integer **N**, which is the number of rooms in your spaceship. The rooms are numbered from 1 to **N**. The following **N** lines each contain a string telling the color of the rooms, from room 1 to room **N**. The strings only contain characters `a-z` (the lower-case English letters) and `0-9` (the number 0 to 9), and the length of each string will be less than or equal to 2.

The next line in the test case is an integer **M**, which indicates the number of turbolifts in your spaceship. The following **M** lines each contain 3 space-separated integers $\mathbf{a_i}$, $\mathbf{b_i}$, $\mathbf{t_i}$, telling us that there is a turbolift that can transport soldiers from room $\mathbf{a_i}$ to room $\mathbf{b_i}$ in $\mathbf{t_i}$ seconds.

The next line in the test case contains an integer **S**, which is the number of soldiers at your command. The following **S** lines each contain two integers: the location and destination of one soldier, $\mathbf{p_j}$ and $\mathbf{q_j}$.

Output

For each test case, output one line containing only the string "Case #x:", where x is the number of the test case (starting from 1). On the next **S** lines, output a single integer: on line **j**, the smallest number of seconds it could take for a soldier to travel from $\mathbf{p_j}$ to $\mathbf{q_j}$. If there is no path from $\mathbf{p_j}$ to $\mathbf{q_j}$, the integer you output should be -1.

Limits

$1 \le \mathbf{S} \le 100$.
$1 \le \mathbf{a_i}, \mathbf{b_i} \le \mathbf{N}$.
$0 \le \mathbf{t_i} \le 1000$.
$1 \le \mathbf{p_j}, \mathbf{q_j} \le \mathbf{N}$.

Small dataset

$1 \le \mathbf{T} \le 10$.
$1 \le \mathbf{N} \le 1000$.
$0 \le \mathbf{M} \le 3000$.

Large dataset

$\mathbf{T} = 1$.
$1 \le \mathbf{N} \le 80000$.

$0 \leq \mathbf{M} \leq 3000$.

Sample

Input	Output

<pre>3
3
gl
t3
t3
3
1 2 217
3 2 567
1 1 21
2
2 1
2 3
4
ca
bl
bl
8z
0
3
1 2
2 3
1 1
8
re
b7
ye
gr
0l
0l
ye
b7
7
4 1 19
2 4 21
2 5 317
4 5 34
4 7 3
4 8 265
8 6 71
3
4 3
2 6
1 4</pre>
	<pre>Case #1:
	-1
	0
	Case #2:
	-1
	0
	0
	Case #3:
	3
	55
	-1</pre>