Selection Contest for HKU ICPC/CCPC Team 2019

The University of Hong Kong

September 12, 2019

Important Notes

- The contest begins at **7:00 pm** and lasts for **150 minutes**, ending at **9:30 pm**.
- The problem set consists of **6 problems**. The problems may not be ordered by difficulty.
- Printed and written notes are allowed, while internet access is prohibited.
- Please use **standard input/output** (e.g. `scanf`/`printf`, `cin`/`cout` or `System.in`/ `System.out.print`) and do not print anything other than those required in the problem.

Problem A — 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

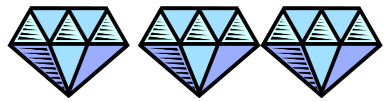
Input will start with a single line containing the number of test cases, between 1 and 100 inclusive. For each test case, there will be a single line giving the number of votes V, $1 \leq V \leq 1000$. Following that line will be V lines, each with a single integer vote between 1 and 1000.

Output

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

Problem B — 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

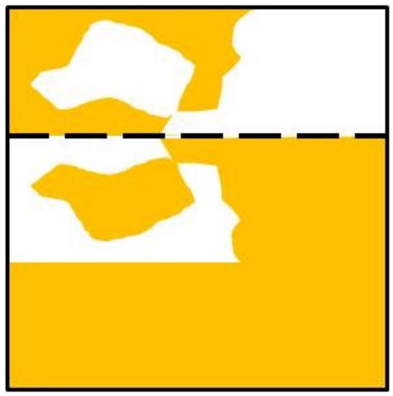
For each test case, output a single line with the length of the longest desirable subsequence of diamonds.

Sample Input	Sample Output
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	2 1 4

Problem C — 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 that 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.

Sample Input

```
3
8 10
#.#..##..#
####..####
###.##....
...#..####
....##....
.#.##..##.
##########
##########
5 20
###########.#.#.#.#.
###########...#.###.
##########..##.#..##
###########..#.#.##.
###########.###...#.
5 5
.####
###.#
##..#
#..##
#####
```

Sample Output

```
3 1 3 10
1 15 5 15
4 1 1 4
```

PROBLEM D — 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, $1 \le T \le 100$, denoting the number of test cases. Each test case begins with a line with a single integer N, $1 \le N \le 100$, 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

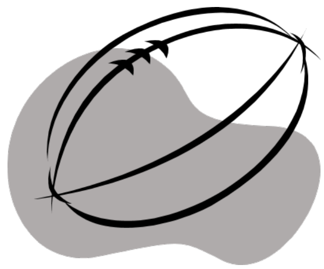
```
4
4
2 1 3 4
4
1 3 2 4
3
1 3 2
6
2 5 1 6 4 3
```

Sample Output

```
Bob
Alice
Bob
Alice
```

Problem E — limit 10 seconds

Top 25



In college football, many different sources create a list of the Top 25 teams in the country. Since it's subjective, these lists often differ, but they're usually very similar. Your job is to compare two of these lists, and determine where they are similar. In particular, you are to partition them into sets, where each set represents the same consecutive range of positions in both lists, and has the same teams, and is as small as possible. If the lists agree completely, you'll have 25 lists (or n, where n is an input). For these lists:

K&R Poll	Lovelace Ranking
A	A
B	C
C	D
D	B
E	E

You'll have 3 sets:

A

B C D

E

Input

The input will start with a single integer on one line giving the number of test cases. There will be at least one but not more than 100 test cases. Each test case will begin with an integer N, $1 \leq N \leq 1{,}000{,}000$, indicating the number of teams ranked. The next N lines will hold the first list, in order. The team names will appear one per line, consist of at most 8 capital letters only. After this will be N lines, in the same format, indicating the second list. Both lists will contain the same team names, and all N team names will be unique.

Output

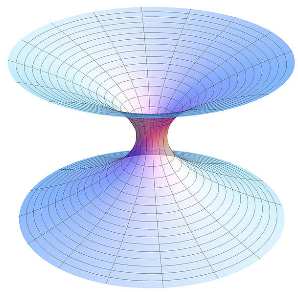
For each test case, simply output the size of each set, in order, on one line, with the numbers separated by a single space. Do not output any extra spaces, and do not output blank lines between numbers.

Sample Input	Sample Output

<pre>3
5
A
B
C
D
E
A
C
D
B
E
3
RED
BLUE
ORANGE
RED
BLUE
ORANGE
3
MOE
LARRY
CURLY
CURLY
MOE
LARRY</pre>
	<pre>1 3 1
	1 1 1
	3</pre>

Problem F — 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 ($0 \leq x, y, z \leq 2 \cdot 10^6$) 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

<pre>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</pre>
	<pre>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.</pre>