1001. A Card Game

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

There are N cards on the table, whose front side is writen one integer number from 1 to M. We call one card "a type k card" if its number is k. The quantity of type i cards is a_i.

Let's play a game with these cards. We divide these cards into M piles by random with the only constrains that the quantity of cards in i-th (indexed from 1) pile must exactly be a_i. The possbility of each card appears in i-th pile is directly proportional to the size of this pile. That is to say, if the size of a pile is A, the possibility for each card appears in this pile is A/N assuming that N is the amount of all cards. We choose pile 1 to start the game. Assuming the we now play this game at pile k, we randomly choose a card from pile k with the same possibility for all cards in it, remember the number written on this card and throw it away. If the number on the chosen card is j, we continue this game at pile j on next round. The game terminates when we are going to get a card from an empty pile.

Now the question is, when the game ends, what is the possibility that all piles are empty?

Input

There is only one input file. The first line is the number of test cases T. T cases follow, each of which contains two lines. The first line is an integer M (1 <= M <= 100), the number of type of cards (and the number of piles, they are exactly the same). The second line contains M positive integers not greater than 1000, the i-th number of which is a_i.

Output

For each test case, output the possibility you are required to calculate. Answers are rounded to 6 numbers after the decimal point.(as shown in the sample output)

Sample Input

```
2
1
5
2
1 2
```

Sample Output

```
Case 1: 1.000000
Case 2: 0.333333
```

Author

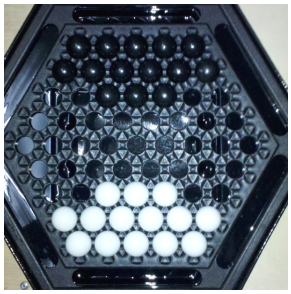
Wywcgs

1002.Abalone

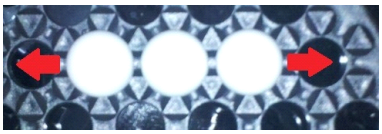
Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

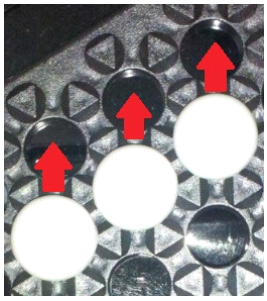
Abalone is a popular board game, the rules are as follow:
1.At first, two players each have 14 marbles, starting to play in turn.



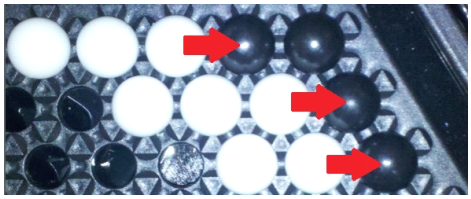
2.For each turn, player must move a line of one, two or three marbles unless he couldn't move any marbles. The move can be either *in-line* (parallel to the line of marbles)



... or broadside (not parallel to the line of marbles), as illustrated below.



3.A player can push their opponent's marbles which are in an adjacent space to their own with an in-line move only. They can only push if the pushing line has more marbles than the pushed line (three can push two or one; two can push one). Marbles must be pushed into an open space or out of board(i.e. not blocked by a marble of either colour).



4.The winner is the first player to push six of the opponent's marbles off of the edge of the board.

Now, it's White player's turn to move, could White player win in this turn? If not, Black player moves, could Black player win in next turn no matter which move White player choose?

Input

The first line is a number T(1<=T<=30), which represents the number of case. The next T blocks follow each indicates a case.
A block has nine lines. They are the state of the situation.
'B' indicate the Black marble.
'W' indicate the White marble.
'.' indicate blank.
The number of marbles for each color will be no bigger than 14 and no smaller than 9.

Output

For each case, output the number of case with "White" or "Draw" or "Black".(as shown in the sample output)

Sample Input

```
3
    B B B B B
   B B B B B B
  . . B B B . .
 . . . . . . . . .
. . . . . . . . . .
 . . . . . . . . .
  . . W W W . .
   W W W W W W
    W W W W W

    . . . . .
   B B B B B B
  . W B B B . .
 . . W . . . . .
. . . W . . . . .
 . . . . . . . . .
  . . . . . . . .
   W W W W W W
    W W W W W

    . . . . .
   W B B B . .
  W W B B B . .
 W . W B B B . .
W W W W . . . . .
 . . . . . . . . .
  . . . . . . . .
   . . . . . .
    . . . . .
```

Sample Output

```
Case 1: Draw
Case 2: White
Case 3: Black
```

Author

NotOnlySuccess

1003. Aircraft

Time Limit: 10000/3000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

You are playing a flying game.
In the game, player controls an aircraft in a 2D-space.
The mission is to drive the craft from starting point to terminal point.
The craft needs wireless signal to move.
A number of devices are placed in the 2D-space, spreading signal.
For a device Di, it has a signal radius -- Ri.
When the distance between the craft and Di is shorter or equal to Ri, it(the craft) gets Di's wireless signal.
Now you need to tell me the shortest path from starting point to terminal point.

Input

The first line of the input file is a single integer T.
The rest of the test file contains T blocks.
Each block starts with an integer n, followed by n devices given as (xi, yi, Ri).
(xi, yi) is position of Di, and Ri is the radius of its signal range.
The first point is the starting point.
The last point is the terminal point.
T <= 25;
2 <= n <= 20 for most cases;
20 < n <= 25 for several cases, completely random generated.
-1000 <= xi, yi <= 1000 , 1 <= ri <= 1000.
All are integers.

Output

For each case, Output "No such path." if the craft can't get to the terminal point.
Otherwise, output a float number, correct the result to 4 decimal places.(as shown in the sample output)

Sample Input

```
2
2
0 0 1
2 0 1
2
0 0 1
4 1 2
```

Sample Output

```
Case 1: 2.0000
Case 2: No such path.
```

Author

MadFroG

1004. Carcassonne

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Carcassonne is a tile-based board game for two to five players. Square tiles are printed by city segments,road segments and field segments.

The rule of the game is to put the tiles alternately. Two tiles share one edge should exactly connect to each other, that is, city segments should be linked to city segments, road to road, and field to field.

To simplify the problem, we only consider putting tiles:
Given n*m tiles. You can rotate each tile, but not flip top to bottom, and not change their order.
How many ways could you rotate them to make them follow the rules mentioned above?

Input

The first line is a number T(1<=T<=50), represents the number of case. The next T blocks follow each indicates a case.
Each case starts with two number N,M(0<N,M<=12)
Then N*M lines follow,each line contains four character clockwise.
'C' indicate City.
'R' indicate Road.
'F' indicate Field.

Output

For each case, output the number of ways mod 1,000,000,007.(as shown in the sample output)

Sample Input

```
3
1 1
RRRR
1 2
RRRF FCCC
8 8
GCGG RRGC GRCR GRGR RCCR GGCC RRGG CRGR
GRRC GRGR CCCR GCGC CRRC CRRR GRCR GRGR
RRCR GRRR CCCR GGGC RRGG RGCR CCGG GCCC
CGCG RRGG CRGR GGRR GRRG CCRR GGGC CRRG
CGRR GGGG GGGG RRGG RRRR RCRR GGCC RGRG
RRCG GRGR GRRR GRGR RCCR RCCC CGGC RGRG
CGCG GRGG RRGG GGGG CGGG CGGG GRGG RGRR
CCRR GCGC GCCC GCCC GGCC GCCG GGCC RGRG
```

Sample Output

```
Case 1: 4
Case 2: 1
Case 3: 1048576
```

Author

NotOnlySuccess

1005. Catan

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

The Settles of Catan is a multiplayer board game. It was one of the first German-style board games to achieve popularity outside of Europe.
The players in the game represent settlers establishing colonies on the island of Catan.Players build settlements, cities, and roads as they settle on the island. Certainly roads can connect the settlements and cities.Players build with resources such as brick, lumber, wool, grain, and ore.
Now look at the board below, do you see the hexagons tiles?

There are six kinds of tiles. Five resources are mentioned above, and one is unproductive desert.
At the beginning of a game, the hexagons tiles will be randomly settled on the board, and each tile will be randomly assigned an integer ranging from 2 to 12. The desert will always be 7.
Every turn a player rolls two six-sided dices, determining which tiles produce resources.

To simplify the problem, we consider the construction phase at the beginning of the game, each player builds two settlements in this phase:
1.We have four players identified by 1,2,3,4.
2.On the first construction phase player 1,2,3,4 take turns to build a settlement separately.
  On the second construction phase player 4,3,2,1 take turns to build a settlement separately.
3.Settlements can only be built on corners of the tiles, see the picture below, you are allowed to build settlements on the 54 corners signed with 0~53.

4.Any two settlements couldn't build adjacently, that means if a settlement have been built on the corner 19, then you couldn't build another settlement on 9, 18 or 20.
5.The probability of the number rolled by two dices are different. For example, if we roll two dices 36 times, the expectation to get 6 or 8 is 5 times, while the expectation to get 2 or 12 is 1.
6.To get better development,you should try you best to get all five kinds of resources.
7.Other players are greedy, when they turn to build settlement, they will choose the corners with highest expectation (if more than one such corners, they will choose the one with minimum sign).

Now as player 1, you should choose the best strategy for building settlement to make the expectation as high as possible.

Input

The first line is a number T(1<=T<=50), represents the number of case. The next T blocks follow each indicates a case.
Five lines follow, indicate the situation.
'B' indicate the Brick.
'W' indicate the Wool.
'L' indicate the Lumber.
'G' indicate the Grain.
'O' indicate the Ore.
'D' indicate the Desert.
Then five lines follow.
Each number indication the number sign of this tile.
There is an empty line between cases.

Output

For each case, If you couldn't get all of five resource,then output -1, else output the highest expectation you could get.(as shown in the sample output)

Sample Input

2

```
  O B B
 G W G W
O L L G G
 L D O B
  B O W
  8 10 9
 3 4 5 12
6 9 11 6 11
 2 7 3 4
  5 10 8

  B W L
 D W L B
L D G B D
 L O D W
  D G W
  8 3 8
 7 2 8 6
9 7 9 8 7
 9 3 7 6
  7 3 8
```

Sample Output

```
Case 1: 19
Case 2: -1
```

Author

NotOnlySuccess

1006. Random Sequence

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/65536 K (Java/Others)**

Problem Description

There is a random sequence L whose element are all random numbers either -1 or 1 with the same possibility. Now we define MAVS, the abbreviate of Maximum Absolute Value Subsequence, to be any (if more than one) subsequences of L whose absolute value is maximum among all subsequences. Given the length of L, your task is to find the expectation of the absolute value of MAVS.

Input

There is only one input file. The first line is the number of test cases T. T positive integers follow, each of which contains one positive number not greater than 1500 denoted the length of L.

Output

For each test case, output the expectation you are required to calculate. Answers are rounded to 6 numbers after the decimal point.(as shown in the sample output)

Sample Input

```
3
1
5
10
```

Sample Output

```
Case 1: 1.000000
Case 2: 2.750000
Case 3: 4.167969
```

Author

wywcgs

1007. Random Maze

Time Limit: 10000/3000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

In the game “A Chinese Ghost Story”, there are many random mazes which have some characteristic:
1.There is only one entrance and one exit.
2.All the road in the maze are unidirectional.
3.For the entrance, its out-degree = its in-degree + 1.
4.For the exit, its in-degree = its out-degree + 1.
5.For other node except entrance and exit, its out-degree = its in-degree.

There is an directed graph, your task is removing some edge so that it becomes a random maze. For every edge in the graph, there are two values a and b, if you remove the edge, you should cost b, otherwise cost a.
Now, give you the information of the graph, your task if tell me the minimum cost should pay to make it becomes a random maze.

Input

The first line of the input file is a single integer T(1<=T<=100).
The rest of the test file contains T blocks.
For each test case, there is a line with four integers, n, m, s and t, means that there are n nodes and m edges, s is the entrance’s index, and t is the exit’s index. Then m lines follow, each line consists of four integers, u, v, a and b, means that there is an edge from u to v.
2<=n<=100, 1<=m<=2000, 1<=s, t<=n, s != t. 1<=u, v<=n. 1<=a, b<=100000

Output

For each case, if it is impossible to work out the random maze, just output the word “impossible”, otherwise output the minimum cost.(as shown in the sample output)

Sample Input

```
2
2 1 1 2
2 1 2 3
5 6 1 4
1 2 3 1
2 5 4 5
5 3 2 3
3 2 6 7
2 4 7 6
3 4 10 5
```

Sample Output

```
Case 1: impossible
Case 2: 27
```

Author

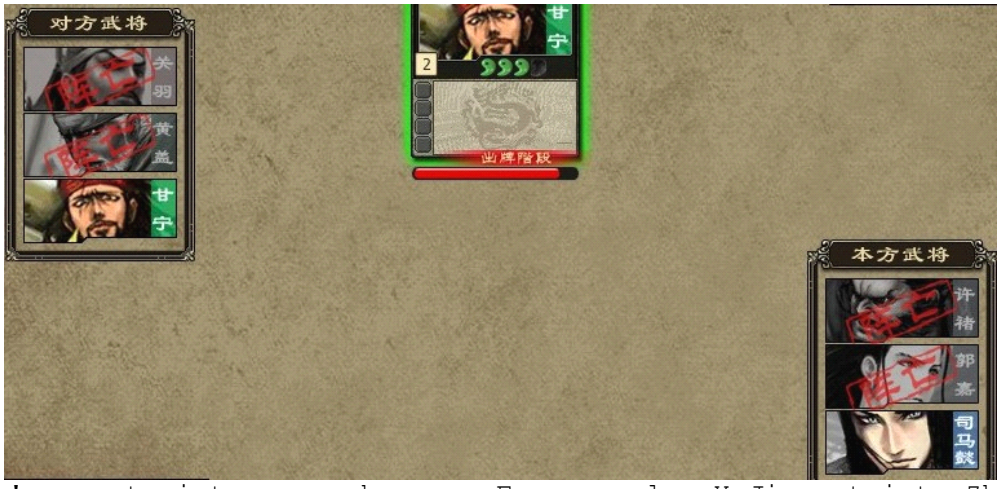
Starvae

1008. SanguoSHA

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Sanguosha has a singled version. Two players each select N heroes and start fighting. If a hero dies, the next follows. If one player's heroes are all dead, he loses.



There're restraints among heroes. For example, YuJi restricts Zhu Geliang, LuXun restricts DaQiao, ZhangJiao restricts MaChao, WeiYan restricts XiaoQiao.
Today I play with friends. I know the heroes and the restraints.(If opponent's hero restraint my hero, my hero will be beaten, others my hero will beat opponent's hero)
Can you arrange my heroes' order,no matter what order of opponent's heroes, so that I can win the game?

Input

The first line is a number T(1<=T<=50), represents the number of case. The next T blocks follow each indicates a case.
The first line is N(3<=N<=6).
The second line has N names(shorter than 20 letter).
The following N lines each contains a restraints. Restraints are given as “k b1 b2 … bk”, which means the opponent's hero restricts my hero b1, b2 … bk. (0<=K<=N)

Output

For each case, first line output the number of case with "Yes" or "No". If Yes, output the order of your heroes separate by a space. If there are more than one order, please output the one with minimum lexicographic order.(as shown in the sample output)

Sample Input

```
2
3
ZhugeLiang HuangYueying ZhenJi
1 ZhugeLiang
```

```
2 HuangYueying ZhenJi
2 ZhugeLiang ZhenJi
4
MaChao YanLiangWenChou YuJin XiaoQiao
2 MaChao XiaoQiao
2 YanLiangWenChou YuJin
1 XiaoQiao
0
```

Sample Output

```
Case 1: No
Case 2: Yes
MaChao YanLiangWenChou XiaoQiao YuJin
```

Author

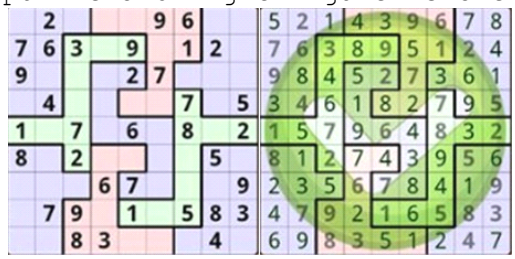
NotOnlySuccess

1009.Squiggly Sudoku

Time Limit: 4000/2000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Today we play a squiggly sudoku, The objective is to fill a 9*9 grid with digits so that each column, each row, and each of the nine Connecting-sub-grids that compose the grid contains all of the digits from 1 to 9.
Left figure is the puzzle and right figure is one solution.



Now, give you the information of the puzzle, please tell me is there no solution or multiple solution or one solution.

Input

The first line is a number T(1<=T<=2500), represents the number of case. The next T blocks follow each indicates a case.
Each case contains nine lines, Each line contains nine integers.
Each module number tells the information of the gird and is the sum of up to five integers:
0~9: '0' means this gird is empty, '1' - '9' means the gird is already filled in.
16: wall to the up
32: wall to the right
64: wall to the down
128: wall to the left
I promise there must be nine Connecting-sub-grids, and each contains nine girds.

Output

For each case, if there are Multiple Solutions or no solution just output "Multiple Solutions" or "No solution". Else output the exclusive solution.(as shown in the sample output)

Sample Input

```
3
144 18 112 208 80 25 54 144 48
135 38 147 80 121 128 97 130 32
137 32 160 144 114 167 208 0 32
192 100 160 160 208 96 183 192 101
209 80 39 192 86 48 136 80 114
152 48 226 144 112 160 160 149 48
128 0 112 166 215 96 160 128 41
128 39 153 32 209 80 101 136 35
192 96 200 67 80 112 208 68 96
```

```
144 48 144 81 81 16 53 144 48
128 96 224 144 48 128 103 128 38
163 208 80 0 37 224 209 0 32
135 48 176 192 64 112 176 192 104
192 101 128 89 80 82 32 150 48
149 48 224 208 16 48 224 192 33
128 0 114 176 135 0 80 112 169
137 32 148 32 192 96 176 144 32
192 96 193 64 80 80 96 192 96

144 88 48 217 16 16 80 112 176
224 176 129 48 128 40 208 16 37
145 32 128 96 196 96 176 136 32
192 32 227 176 144 80 96 192 32
176 192 80 98 160 145 80 48 224
128 48 144 80 96 224 183 128 48
128 36 224 144 51 144 32 128 105
131 64 112 136 32 192 36 224 176
224 208 80 64 64 116 192 83 96
```

Sample Output

```
Case 1:
521439678
763895124
984527361
346182795
157964832
812743956
235678419
479216583
698351247
Case 2:
No solution
Case 3:
Multiple Solutions
```

Author

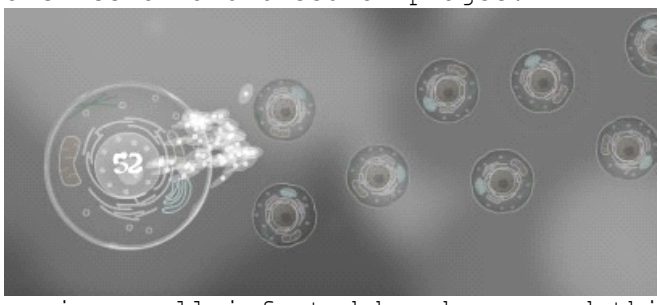
NotOnlySuccess

1010.War

Time Limit: 3000/1000 MS (Java/Others) **Memory Limit: 65536/32768 K (Java/Others)**

Problem Description

Phage War is a little flash game. In this game, we want infect all cells by the transmission and breed of phages.



Originally, there is a cell infected by phages and this cell can breed a new phage every second. You should know that only the new born phages can inject other cells.

There are n cells around this cell, numbered from 1 to n. If there are Di phages reaching the i-th cell, the cell would be infected, and the phages journey will cost Ti seconds. To simplify it, we assume these phages will stay in this new cell and they can't infect other cells. And the new cell cannot breed new phages and infect other cells.
Can you tell me how much time it costs to infect all cells at least?

Input

In the first line there is an integer T (T <= 50), indicates the number of test cases.
In each case, the first line contains a integers N (1 <= N <= 10^5). Then there are N lines, each line contain two integers Di, Ti (1<=Di, Ti<=100).

Output

For each case, output the least time needed in one line.

Sample Input

```
2
2
2 1
5 6
2
1 11
3 10
```

Sample Output

```
Case 1: 11
Case 2: 14
```

Author

Ambition