Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION – 1

A **tree** consists of a node and some (zero, one or two) subtrees connected to it. These subtrees are called children.

A **specification** of the tree is a sequence of digits. If the number of children in the tree is:

- zero, then the specification is a sequence with only one element '0';
- one, the specification begins with '1' followed by the specification of the child;
- two, the specification begins with '2' followed by the specification of the first child, and then by the specification of the second child.

Each of the vertices in the tree must be painted either red or green or blue.
However, we need to obey the following rules:

- the vertex and its child cannot have the same color,
- if a vertex has two children, then they must have different colors.

How many vertices may be painted green?

Task

Write a program which:

- reads the specification of the tree from the standard input,
- computes the maximal and the minimal number of vertices that may be painted green,
- writes the results in the standard output.

Input

The number of test cases t is in the first line of input, then t test cases follow separated by an empty line. Each test case consists of one word (no longer then 10000 characters), which is a specification of a tree.

Output

Your program should write for each test case exactly two integers separated by a single space, which respectively denote the maximal and the minimal number of vertices that may be painted green.

Example

Sample input:
```
1
1122002010
```

Sample output:
```
5 2
```

QUESTION – 2

Vonny loves playing with dominos. And so she owns a standard set of dominos. A standard set of dominos consists of 28 pieces called bones, tiles or stones. Each bone is a rectangular tile with a line dividing its face into two square ends. Each square is labeled with a number between 0 and 6. The 28 stones are labeled (0,0),(0,1),(0,2),(0,3),(0,4),(0,5),(0,6), (1,1),(1,2),...,(5,5),(5,6),(6,6). Tommy - the brother of Vonny - build a box for Vonny's dominos. This box is sized 7 x 8 squares. Every square is labeled with a number between 0 and 6. You can see a example box here.

```
0 3 0 2 2 0 2 3
1 5 6 5 5 1 2 2
3 4 1 4 5 4 4 4
6 6 1 0 5 2 3 0
4 0 3 2 4 1 6 0
1 4 1 5 6 6 3 0
1 2 6 5 5 6 3 3
```

Now Vonny wants to arrange her 28 stones in such way that her stones cover all squares of the box. A stone can only be placed on two adjacent squares if the numbers of the squares and of the domino stone are equal. Tommy asks Vonny in how many different ways she can arrange the dominos. Tommy assumes that Vonny need a lot of time to answer the question. And so he can take some of Vonny's candies while she solves the task. But Vonny is a smart and clever girl. She asks you to solve the task and keeps an eye on her candies.

Input

The first line of the input contains the number of testcases. Each case consists of 56 numbers (7 rows and 8 cols) between 0 and 6 which represents Tommy's box.

Output

For each testcase output a single line with the number which answers Tommy's question.

Example

Input:

```
2
0 3 0 2 2 0 2 3
1 5 6 5 5 1 2 2
3 4 1 4 5 4 4 4
6 6 1 0 5 2 3 0
4 0 3 2 4 1 6 0
1 4 1 5 6 6 3 0
1 2 6 5 5 6 3 3

5 3 1 0 0 1 6 3
0 2 0 4 1 2 5 2
1 5 3 5 6 4 6 4
0 5 0 2 0 4 6 2
4 5 3 6 0 6 1 1
2 3 5 3 4 4 5 3
```

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

Output:

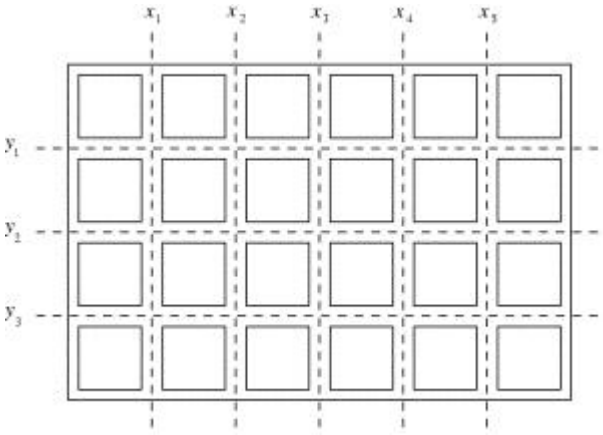
```
18
1
```

QUESTION – 3

We are given a bar of chocolate composed of *m*n* square pieces. One should break the chocolate into single squares. Parts of the chocolate may be broken along the vertical and horizontal lines as indicated by the broken lines in the picture.

A single break of a part of the chocolate along a chosen vertical or horizontal line divides that part into two smaller ones. Each break of a part of the chocolate is charged a cost expressed by a positive integer. This cost does not depend on the size of the part that is being broken but only depends on the line the break goes along. Let us denote the costs of breaking along consecutive vertical lines with $x_1, x_2, ..., x_{m-1}$ and along horizontal lines with $y_1, y_2, ..., y_{n-1}$.

The cost of breaking the whole bar into single squares is the sum of the successive breaks. One should compute the minimal cost of breaking the whole chocolate into single squares.



For example, if we break the chocolate presented in the picture first along the horizontal lines, and next each obtained part along vertical lines then the cost of that breaking will be $y_1+y_2+y_3+4*(x_1+x_2+x_3+x_4+x_5)$.

Task

Write a program that for each test case:

- Reads the numbers $x_1, x_2, ..., x_{m-1}$ and $y_1, y_2, ..., y_{n-1}$
- Computes the minimal cost of breaking the whole chocolate into single squares, writes the result.

Input

One integer in the first line, stating the number of test cases, followed by a blank line. There will be not more than 20 tests.

For each test case, at the first line there are two positive integers *m* and *n* separated by a single space, 2 <= *m*,*n* <= 1000. In the successive *m*-1 lines there are numbers x_1, x_2, ..., x_{m-1}, one per line, 1 <= x_i <= 1000. In the successive *n*-1 lines there are numbers y_1, y_2, ..., y_{n-1}, one per line, 1 <= y_i <= 1000.

The test cases will be separated by a single blank line.

Output

For each test case : write one integer - the minimal cost of breaking the whole chocolate into single squares.

Example

Input:

```
1

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

Output:

```
42
```

QUESTION – 4

Rain has pummeled on the cows' field, a rectangular grid of R rows and C columns (1 <= R <= 50, 1 <= C <= 50). While good for the grass, the rain makes some patches of bare earth quite muddy. The cows, being meticulous grazers, don't want to get their hooves dirty while they eat.

To prevent those muddy hooves, Farmer John will place a number of wooden boards over the muddy parts of the cows' field. Each of the boards is 1 unit wide, and can be any length long. Each board must be aligned parallel to one of the sides of the field.

Farmer John wishes to minimize the number of boards needed to cover the muddy spots, some of which might require more than one board to cover. The boards may not cover any grass and deprive the cows of grazing area but they can overlap each other.

Compute the minimum number of boards FJ requires to cover all the mud in the field.

Input

t – the number of test cases, then *t* test cases follows.
Each test case is of the following form:
Two space-separated integers: *R* and *C*, then R lines follows
Each line contains a string of C characters, with '*' representing a muddy patch, and '.' representing a grassy patch. No spaces are present.

Output

For each test case output a single integer representing the number of boards FJ needs.

Example

Input:
```
1
4 4
*.*.
.***
***.
..*.
```

Output:
```
4
```

Output details:
```
Boards 1, 2, 3 and 4 are placed as follows:
1.2.
.333
444.
..2.
Board 2 overlaps boards 3 and 4.
```

QUESTION – 5

Alibaba the famous character of our childhood stories would like to be immortal in order to keep bringing happiness to children. In order to reach this status he needs to prove that he is still able to do some unusual things. There are **n** treasures, (**n** <= 10000) each in a different place located along a straight road. Each treasure has a time limit, after that it vanishes. Alibaba must take all the **n** treasures, and he must do it quickly. So he needs to figure out the order in which he should take the treasures before their deadlines starting from the most favorable position. Alibaba has the list of places and deadlines of the treasures. A place **i** is located at distance $\mathbf{d_i}$ from the leftmost end of the road. The time it takes to take a treasure is instantaneous.

Alibaba must find **the smallest time** by which he can take all the treasures.

Input

The first line of the input contains an integer **K** <= 10 - determining the number of datasets

Each data set in the input stands for a particular set of treasures. For each set of treasures the input contains the number of treasures, and the list of pairs place - deadline in increasing order of the locations. White spaces can occur freely between the numbers in the input.The input data are correct.

Output

For each set of data the program prints the result to the standard output on a separate line. The solution is represented by the smallest time by which Alibaba can take all the treasures before they vanish. If this is not possible then the output is "No solution".

Example

Input:

```
2
5
1 3
3 1
5 8
8 19
10 15
5
1 5
2 1
3 4
4 2
5 3
```

Output:

```
11
No solution
```