

Kickstart Round B 2017

A. Math Encoder[B. Center](#)[C. Christmas Tree](#)[Questions asked](#)

Submissions

Math Encoder

7pt	Not attempted 820/920 users correct (89%)
16pt	Not attempted 411/802 users correct (51%)

Center

13pt	Not attempted 190/437 users correct (43%)
21pt	Not attempted 97/175 users correct (55%)

Christmas Tree

11pt	Not attempted 442/522 users correct (85%)
32pt	Not attempted 59/199 users correct (30%)

Top Scores

azure97	100
pps789	100
saffahyjp	100
wifi	100
Uhateme	100
BangBangBang	100
Tian.Xie	100
pwypeanut	100
mengrao	100
Doju	100

Problem A. Math Encoder

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
7 points

[Solve A-small](#)

Large input
16 points

[Solve A-large](#)**Problem**

Professor Math is working on a secret project and is facing a challenge where a list of numbers need to be encoded into a single number in the most efficient manner. After much research, Professor Math finds a 3 step process that can best encode the numbers:

1. The first step is to find all possible non-empty subsets of the list of numbers and then, for each subset, find the difference between the largest and smallest numbers (that is, the largest minus the smallest) in that subset. Note that if there is only one number in a subset, it is both the largest and the smallest number in that subset. The complete set itself is also considered a subset.
2. Then add up all the differences to get the final encoded number.
3. As the number may be large, output the number modulo $10^9 + 7$ (1000000007).

The professor has shared an example and its explanation below. Given a list of numbers, can you help the professor build an efficient function to compute the final encoded number?

Input

The first line of the input gives the number of test cases, **T**. This is followed by T test cases where each test case is defined by 2 lines:

1. The first line gives a positive number **N**: the number of numbers in the list and
2. The second line contains a list of **N** positive integers **K_i**, sorted in non-decreasing order.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the final encoded number.

Since the output can be a really big number, we only ask you to output the remainder of dividing the result by the prime $10^9 + 7$ (1000000007).

Limits

$1 \leq T \leq 25$.
 $1 \leq K_i \leq 10000$, for all i.
 $K_i \leq K_{i+1}$, for all $i < N - 1$.

Small dataset

$1 \leq N \leq 10$.

Large dataset

$1 \leq N \leq 10000$.

Sample

Input	Output
1	Case #1: 44
4	
3 6 7 9	

Explanation for the sample input

1. Find all subsets and get the difference between largest & smallest numbers:
[3], largest-smallest = 3 - 3 = 0.
[6], largest-smallest = 6 - 6 = 0.

- [7], largest-smallest = $7 - 7 = 0$.
 - [9], largest-smallest = $9 - 9 = 0$.
 - [3, 6], largest-smallest = $6 - 3 = 3$.
 - [3, 7], largest-smallest = $7 - 3 = 4$.
 - [3, 9], largest-smallest = $9 - 3 = 6$.
 - [6, 7], largest-smallest = $7 - 6 = 1$.
 - [6, 9], largest-smallest = $9 - 6 = 3$.
 - [7, 9], largest-smallest = $9 - 7 = 2$.
 - [3, 6, 7], largest-smallest = $7 - 3 = 4$.
 - [3, 6, 9], largest-smallest = $9 - 3 = 6$.
 - [3, 7, 9], largest-smallest = $9 - 3 = 6$.
 - [6, 7, 9], largest-smallest = $9 - 6 = 3$.
 - [3, 6, 7, 9], largest-smallest = $9 - 3 = 6$.
2. Find the sum of the differences calculated in the previous step:
 $3+4+6+1+3+2+4+6+6+3+6$
 $= 44$.
3. Find the answer modulo $10^9 + 7$ (1000000007):
 $44 \% 1000000007 = 44$

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform

Kickstart Round B 2017

[A. Math Encoder](#)**B. Center**[C. Christmas Tree](#)[Questions asked](#)

Submissions

Math Encoder

7pt	Not attempted 820/920 users correct (89%)
16pt	Not attempted 411/802 users correct (51%)

Center

13pt	Not attempted 190/437 users correct (43%)
21pt	Not attempted 97/175 users correct (55%)

Christmas Tree

11pt	Not attempted 442/522 users correct (85%)
32pt	Not attempted 59/199 users correct (30%)

Top Scores

azure97	100
pps789	100
saffahyjp	100
wifi	100
Uhateme	100
BangBangBang	100
Tian.Xie	100
pwypeanut	100
mengrao	100
Doju	100

Problem B. Center

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
13 points

Solve B-small

Large input
21 points

Solve B-large

Problem

There are N weighted points in a plane. Point i is at (X_i, Y_i) and has weight W_i .

In this problem, we need to find a special center of these points. The center is a point (X, Y) such that the sum of $\max(|X - X_i|, |Y - Y_i|) * W_i$ is minimum.

Input

The input starts with one line containing exactly one integer T , which is the number of test cases. T test cases follow.

Each test case begins with one line containing one integer N . N lines follow. Each line contains three space-separated real numbers X_i , Y_i , and W_i . X_i , Y_i and W_i have exactly 2 digits after the decimal point.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the sum of $\max(|X - X_i|, |Y - Y_i|) * W_i$ for center (X, Y) .

y will be considered correct if it is within an absolute or relative error of 10^{-6} of the correct answer. See the [FAQ](#) for an explanation of what that means, and what formats of real numbers we accept.

Limits

$1 \leq T \leq 10$.
 $-1000.00 \leq X_i \leq 1000.00$.
 $-1000.00 \leq Y_i \leq 1000.00$.

Small dataset

$1 \leq N \leq 100$;
 $W_i = 1.0$, for all i .

Large dataset

$1 \leq N \leq 10000$;
 $1.0 \leq W_i \leq 1000.0$, for all i .

Sample

Input	Output
3	Case #1: 1.0
2	Case #2: 4.0
0.00 0.00 1.00	Case #3: 1.0
1.00 0.00 1.00	
4	
1.00 1.00 1.00	
1.00 -1.00 1.00	
-1.00 1.00 1.00	
-1.00 -1.00 1.00	
2	
0.00 0.00 1.00	
1.00 0.00 2.00	

Powered by



Google Cloud Platform

Kickstart Round B 2017

[A. Math Encoder](#)[B. Center](#)**C. Christmas Tree**[Questions asked](#)

Submissions

Math Encoder

7pt	Not attempted 820/920 users correct (89%)
16pt	Not attempted 411/802 users correct (51%)

Center

13pt	Not attempted 190/437 users correct (43%)
21pt	Not attempted 97/175 users correct (55%)

Christmas Tree

11pt	Not attempted 442/522 users correct (85%)
32pt	Not attempted 59/199 users correct (30%)

Top Scores

azure97	100
pps789	100
saffahyjp	100
wifi	100
Uhateme	100
BangBangBang	100
Tian.Xie	100
pwypeanut	100
mengrao	100
Doju	100

Problem C. Christmas 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
11 points

Solve C-small

Large input
32 points

Solve C-large

Problem

You are given a rectangular grid with **N** rows and **M** columns. Each cell of this grid is painted with one of two colors: green and white. Your task is to find the number of green cells in the largest Christmas tree in this grid.

To define a Christmas tree, first we define a **good triangle** as follows:

A good triangle with top point at row **R**, column **C** and height **h** ($h > 0$) is an isosceles triangle consisting entirely of green cells and pointing upward. Formally, this means that: The cell (**R**, **C**) is green, and for each **i** from 0 to **h-1** inclusive, the cells in row **R+i** from column **C-i** to column **C+i** are all green.

For example:

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

is a good triangle with height 3. The # cells are green and the . cells are white. Note that there is a green cell that is not part of the good triangle, even though it touches the good triangle.

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

is **NOT** a good triangle with height 3, because the 5th cell in the 3rd row is white. However, there are good triangles with height 2 present.

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

is **NOT** a good triangle with height 3. However, there are good triangles with height 2 present.

A **K**-Christmas tree is defined as follows:

- It contains exactly **K** good triangles in vertical arrangement.
- The top cell of the **i+1**-th triangle must share its top edge with the bottom edge of any one of the cells at the base of the **i**-th triangle. This means that, if the base of the **i**-th triangle is at row **r**, from column **c1** to column **c2**, then the top of the **i+1**-th triangle must be on row **r+1**, in a column somewhere between **c1** and **c2**, inclusive.

For example, if **K** = 2:

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

is a valid 2-Christmas tree. Note that the height of the 2 good triangles can be different.

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

is also a valid 2-Christmas tree. Note that a good triangle can be of height 1 and have only one green cell.

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

The two good triangles with height 3 does **NOT** form a valid 2-Christmas tree, because the 2nd triangle must start from the 4-th row.

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

The two good triangles with height 2 does **NOT** form a valid 2-Christmas tree, because the top of the 2nd triangle must be in a column between 3 and 5, inclusive.

You need to find the K-Christmas tree with the largest number of green cells.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of three lines:

- The first line contains 3 space-separated integers **N**, **M** and **K**, where **N** is the number of rows of the grid, **M** is the number of columns of the grid and **K** is the number of good triangle in the desired Christmas tree.
- The next **N** lines each contain exactly **M** characters. Each character will be either **.** or **#**, representing a white or green cell, respectively.

Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is the number of green cells in the largest K-Christmas tree. If there is no K-Christmas tree, output 0.

Limits

$1 \leq T \leq 100$.

$1 \leq M \leq 100$.

$1 \leq N \leq 100$.

Each cell in the grid is either **.** or **#**.

Small dataset

K = 1.

Large dataset

$1 \leq K \leq 100$.

Sample

Input	Output
4	Case #1: 9
3 5 1	Case #2: 0
..#..	Case #3: 9
.###.	Case #4: 10
#####	
3 5 1	
.....	
.....	
.....	
4 5 1	
#####	
#####	
#####	
#####	
4 5 2	
#####	
#####	
#####	
#####	

In sample case #1, the largest 1-Christmas tree has 9 green cells:

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

In sample case #2, there is no 1-Christmas tree.

In sample case #3, one largest 1-Christmas tree with 9 green cells is:

```
#####
#####
#####
#####
```

In sample case #4, one largest 2-Christmas tree with 10 green cells is:

```
#####
#####
#####
#####
```

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform