

Problem A. Boomerang Constellations

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 512 megabytes

Its price was 10 points on the official contest.

The night sky can be modeled as an infinite 2D plane. There are N stars at distinct positions on this plane, the i th of which is at coordinates (X_i, Y_i) .

A boomerang constellation is a pair of distinct equal-length line segments which share a single endpoint, such that both endpoints of each segment coincide with a star's location.

Two boomerang constellations are distinct if they're not made up of the same unordered pair of line segments. How many distinct boomerang constellations can you spot?

Input

Input begins with an integer T ($1 \leq T \leq 50$, the number of nights on which you look out at the sky. For each night, there is first a line containing the integer N ($1 \leq N \leq 2,000$). Then, N lines follow, the i th of which contains the space-separated integers X_i and Y_i ($-10000 \leq X_i, Y_i \leq 10000$).

Output

For the i -th night, print a line containing "Case #i: " followed by the number of boomerang constellations in the night sky.

Examples

standard input	standard output
5	Case #1: 0
3	Case #2: 4
0 0	Case #3: 4
0 1	Case #4: 3
0 3	Case #5: 12
5	
0 0	
0 1	
0 2	
0 3	
0 4	
4	
0 0	
0 100	
100 0	
100 100	
4	
0 0	
-3 4	
0 5	
-5 0	
6	
5 6	
6 5	
7 6	
6 7	
7 8	
8 7	

Note

On the first night, every pair of stars is a unique distance apart, so there are no boomerang constellations. On the second night,

there are 4 boomerang constellations. One of them consists of the line segments $(0,0)-(0,2)$ and $(0,2)-(0,4)$.

Problem B. High Security

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 512 megabytes

Its price was 25 points on the official contest.

A top-secret algorithmic research facility has decided to up its security by hiring guards to keep watch over the premises. After all, they don't want anyone sneaking in and learning the answers to questions such as "does $P = NP$?" and "what are the solutions to the 2016 Facebook Hacker Cup problems?".

When viewed from above, the facility can be modeled as a grid G with 2 rows and N columns. The j -th cell in the i th row is either empty (represented by $G_{i,j} = "."$) or occupied by a building ($G_{i,j} = "X"$), and the grid includes at least one empty cell.

Guards may be potentially stationed in any of the empty cells. A guard can see not only their own cell, but also all contiguous empty cells in each of the 4 compass directions (up, down, left, and right) until the edge of the grid or a building. For example, in the grid below, the guard ("G") can see every cell marked with an asterisk ("*"):

```
.*.X.X..
*G*****X
```

What is the minimum number of guards required such that every empty cell in the grid can be seen by at least one of them?

Input

Input begins with an integer T ($1 \leq T \leq 200$), the number of facilities that need guarding. For each facility, there is first a line containing the integer N ($1 \leq N \leq 1000$). The next line contains the grid cells $G_{1,1}$ to $G_{1,N}$ in order. The third line contains the grid cells $G_{2,1}$ to $G_{2,N}$ in order.

Output

For the i -th facility, print a line containing "Case #i: " followed by the number of guards required to guard the facility.

Examples

standard input	standard output
5	Case #1: 3
8	Case #2: 3
...X.X..	Case #3: 2
.....X	Case #4: 5
5	Case #5: 6
.X.X.	
.XXX.	
7	
.....X.	
.X.....	
9	
..X.X.X..	
..X...X.X	
15	
.X.....X..	
.X...XX.X.X....	

Note

In the first case, one solution is to place three guards as follows:

.G.X.XG.
....G..X

Problem C. The Price is Correct

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 512 megabytes

Its price was 25 points on the official contest.

You've managed to become a contestant on the hottest new game show, The Price is Correct!

After asking you to come on down to the stage, the show's host presents you with a row of N closed boxes, numbered from 1 to N in order, each containing a secret positive integer. A curtain opens to reveal a shiny, new tricycle — you recognize it as an expensive, top-of-the-line model.

The host then proceeds to explain the rules: you must select a contiguous sequence of the boxes (boxes $a \dots b$, for some $1 \leq a \leq b \leq N$). Your chosen boxes will then be opened, and if the sum of the numbers inside is no greater than the price of the tricycle, you win it!

You'd sure like to win that tricycle. Fortunately, not only are you aware that its price is exactly P , but you've paid off the host to let you in on the contents of the boxes! You know that each box i contains the number B_i .

How many different sequences of boxes can you choose such that you win the tricycle? Each sequence is defined by its starting and ending box indices (a and b).

Input

Input begins with an integer T ($1 \leq T \leq 40$), the number of times you appear on The Price is Correct. For each show, there is first a line containing the space-separated integers N ($1 \leq N \leq 10^5$) and P ($1 \leq P \leq 10^9$). The next line contains N space-separated integers, B_1 through B_N in order ($1 \leq B_i \leq 10^9$).

Output

For the i -th show, print a line containing "Case #i: " followed by the number of box sequences that will win you the tricycle.

Examples

standard input	standard output
5	Case #1: 10
4 50	Case #2: 0
10 10 10 10	Case #3: 3
4 50	Case #4: 9
51 51 51 51	Case #5: 18
3 1000000000	
1000000000 1000000000	
1000000000	
6 6	
1 2 3 4 5 6	
10 77	
12 3 52 25 9 83 45 21	
33 3	

Note

In the first case no sequence adds up to more than 50, so all 10 sequences are winners. In the fourth case, you can select any single box, or the sequences (1,2), (1,3), and (2,3) for 9 total winning sequences.

Problem D. Text Editor

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 512 megabytes

Its price was 40 points on the official contest.

You have a list of N distinct words, consisting of only lowercase letters. You'd like to print any K words from this list, one per page, in any order.

You will accomplish this using a very basic text editor. It supports 3 types of operations: typing a letter, deleting the previous letter, and printing the current document. Note that it does not allow the cursor to be moved! This means that the first operation may only add a letter to the end of the document, and the second may only delete the last letter (if any). Due to issues with memory leakage, you also need to remember to leave the document completely empty after you've printed your K words!

What's the minimum number of operations required to get the job done?

As an example, let's say that you want to print 3 of the following 5 words: **a**, **hair**, **box**, **awesome**, **hail**.

One optimal sequence of 15 operations is as follows:

- type 'h', 'a', 'i', and 'r' (document = 'hair')
- print
- backspace (document = 'hai')
- type 'l' (document = 'hail')
- print
- backspace 4 times (document = empty)
- type 'a' (document = 'a')
- print
- backspace (document = empty)

Input

Input begins with an integer T ($1 \leq T \leq 100$), the number of sets of words you want to print. For each set, there is first a line containing the space-separated integers N and K ($1 \leq K \leq N \leq 300$). The next N lines contain the set of words, one per line. The total length of all N words in each set will be at most 10^5 characters.

Output

Examples

standard input	standard output
5	Case #1: 9
3 3	Case #2: 9
x	Case #3: 11
y	Case #4: 15
z	Case #5: 26
2 1	
loooong	
tiny	
4 3	
fff	
fffff	
ff	
ffff	
5 3	
a	
hair	
box	
awesome	
hail	
8 6	
fox	
of	
xfox	
foo	
foxxx	
off	
foff	
foox	