KICK START

**PROBLEM 0.**

You have gathered NN bags of candy and you want to distribute the candy amongst MM kids. The ii-th bag contains CiCi pieces of candy. You want to make sure that every kid get the same amount of candy and that the number of pieces of candy they receive is the greatest possible. You can open each bag and mix all pieces of candy before distributing them to the kids.

How many pieces of candy will remain after you share the candy amongst kids, based on the rules described above?

### Input

The first line of the input gives the number of test cases, TT. TT test cases follow.

Each test case consists of two lines. The first line of each test case contains two integers: integer NN, the number of candy bags, and MM, the number of kids.

The next line contains NN non-negative integers C1,C2,…,CNC1,C2,…,CN representing array CC, where the ii-th integer represents the number of candies in the ii-th bag.

### Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 11) and y is the number of candies that will remain if you divide candies between kids according to the rules described above.

### Limits

Time limit: 60 seconds.  
Memory limit: 1 GB.

#### Test Set 1

1≤T≤1001≤T≤100.  
1≤N≤1051≤N≤105.  
1≤M≤1041≤M≤104.  
0≤Ci≤10000≤Ci≤1000, for all ii from 11 to NN.

### Sample

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U1JD-w0p8mOA8cnHEYtUwBNYe8SAF2dYgefBNUcUQJJr7CF60SXpYc-YC1ULsaEWqCrvmbVjnx_ut6w0Rlo1FTfmw/sample_problem_sample_ts1_input.txt?dl=1)

content\_copy

2

7 3

1 2 3 4 5 6 7

5 10

7 7 7 7 7

Sample Output

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U243pA-qIzj_8QOGuog9OXvdYrj5bQsjGzYklnKtiAgc6Fxm8pbZZmwJJC8yjwOGiDHH-F3cbmmsiMnAD4mZlgN-oU/sample_problem_sample_ts1_output.txt?dl=1)

content\_copy

Case #1: 1

Case #2: 5

In Sample Case #1, we have N=7N=7 bags of candy. In total we have 1+2+3+4+5+6+7=281+2+3+4+5+6+7=28 candies that we want to divide between M=3M=3 kids. Every kid can get 99 pieces of candy, so 28−3×9=128−3×9=1 pieces of candy will remain.

In Sample Case #2, we have N=5N=5 bags of candy. In total we have 7+7+7+7+7=357+7+7+7+7=35 candies that we want to divide between M=10M=10 kids. Every kid can get 33 pieces of candy, so 35−10×3=535−10×3=5 pieces of candy will remain.

### Sample codes

If you want to learn how to structure your code, read from the standard input, or print to the standard output, please find the sample solutions to this problem below in all the programming languages that are available on our platform.

**PROBLEM 1.**

Isyana is given the number of visitors at her local theme park on NN consecutive days. The number of visitors on the ii-th day is ViVi. A day is *record breaking* if it satisfies both of the following conditions:

* Either it is the first day, or the number of visitors on the day is strictly larger than the number of visitors on each of the previous days.
* Either it is the last day, or the number of visitors on the day is strictly larger than the number of visitors on the following day.

Note that the very first day could be a record breaking day!

Please help Isyana find out the number of record breaking days.

Input

The first line of the input gives the number of test cases, TT. TT test cases follow. Each test case begins with a line containing the integer NN. The second line contains NN integers. The ii-th integer is ViVi and represents the number of visitors on the ii-th day.

Output

For each test case, output one line containing Case #xx: yy, where xx is the test case number (starting from 11) and yy is the number of record breaking days.

Limits

Time limit: 20 seconds.  
Memory limit: 1 GB.  
1≤T≤1001≤T≤100.  
0≤Vi≤2×1050≤Vi≤2×105, for all ii.

Test Set 1

1≤N≤10001≤N≤1000.

Test Set 2

1≤N≤2×1051≤N≤2×105, for at most 10 test cases.  
For the remaining cases, 1≤N≤10001≤N≤1000.

Sample

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U3_rhJARjpe_SaiHzpmtTHvTgoxJtcqmzUkFCdBz3yX8ruywaHshcjA2pAgYnBZYq-yUeMDbNKQfeTrjctw4buWcw/record_breaker_sample_ts1_input.txt?dl=1)

content\_copy

4

8

1 2 0 7 2 0 2 0

6

4 8 15 16 23 42

9

3 1 4 1 5 9 2 6 5

6

9 9 9 9 9 9

Sample Output

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U2CF1EBOauYgyebXBotbpVl1XQQCo-oX_LUv39KRvFeLCQO3qNV_eKhAphPVRw5CiDLQrg-9lFNGzLICUD3MMY7kXc/record_breaker_sample_ts1_output.txt?dl=1)

content\_copy

Case #1: 2

Case #2: 1

Case #3: 3

Case #4: 0

In Sample Case #1, the underlined numbers in the following represent the record breaking days: 12––07––202012\_07\_2020.

In Sample Case #2, only the last day is a record breaking day: 4815162342–––4815162342\_.

In Sample Case #3, the first, the third, and the sixth days are record breaking days: 3––14––159––2653\_14\_159\_265.

Problem 2.

Banny has just bought a new programmable robot. Eager to test his coding skills, he has placed the robot in a grid of squares with RR rows (numbered 11 to RR from north to south) and CC columns (numbered 11 to CC from west to east). The square in row rr and column cc is denoted (r,c)(r,c).

Initially the robot starts in the square (SRSR, SCSC). Banny will give the robot NN instructions. Each instruction is one of N, S, E, or W, instructing the robot to move one square north, south, east, or west respectively.

If the robot moves into a square that it has been in before, the robot will continue moving in the same direction until it reaches a square that it *has not* been in before. Banny will never give the robot an instruction that will cause it to move out of the grid.

Can you help Banny determine which square the robot will finish in, after following the NN instructions?

### Input

The first line of the input gives the number of test cases, TT. TT test cases follow. Each test case starts with a line containing the five integers NN, RR, CC, SRSR, and SCSC, the number of instructions, the number of rows, the number of columns, the robot's starting row, and the robot's starting column, respectively.

Then, another line follows containing a single string consisting of NN characters; the ii-th of these characters is the ii-th instruction Banny gives the robot (one of N, S, E, or W, as described above).

### Output

For each test case, output one line containing Case #xx: rr cc, where xx is the test case number (starting from 1), rr is the row the robot finishes in, and cc is the column the robot finishes in.

### Limits

Memory limit: 1 GB.  
1≤T≤1001≤T≤100.  
1≤R≤5×1041≤R≤5×104.  
1≤C≤5×1041≤C≤5×104.  
1≤SR≤R1≤SR≤R.  
1≤SC≤C1≤SC≤C.  
The instructions will not cause the robot to move out of the grid.

#### Test Set 1

Time limit: 20 seconds.

1≤N≤1001≤N≤100.

#### Test Set 2

Time limit: 60 seconds.

1≤N≤5×1041≤N≤5×104.

### Sample

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U24EfCFmSlQEp0chpULkOK81jPiXVdtyMd7BMSyP0WFygxFJ2RrREJcwYKg2SuDEbb6Is8RvavUZVUPNJtcVA/wiggle_walk_sample_ts1_input.txt?dl=1)

content\_copy

3

5 3 6 2 3

EEWNS

4 3 3 1 1

SESE

11 5 8 3 4

NEESSWWNESE

Sample Output

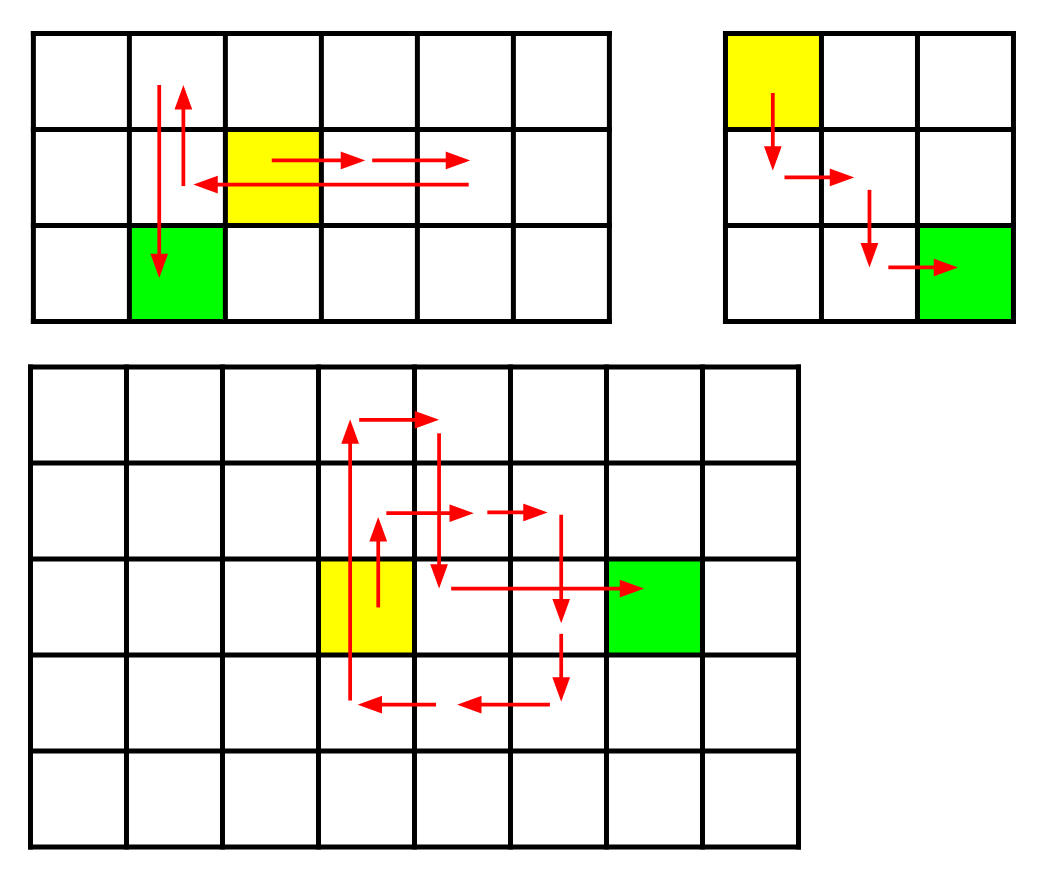
[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U2rzpVTZELtUxPZhsyTpBUHzdJo03yLQ4ik46im74Ot9NXf4modGZMLKqn8iUERBFZRtmibWqQF2BlEjk1Oldg/wiggle_walk_sample_ts1_output.txt?dl=1)

content\_copy

Case #1: 3 2

Case #2: 3 3

Case #3: 3 7

Sample Case #1 corresponds to the top-left diagram, Sample Case #2 corresponds to the top-right diagram, and Sample Case #3 corresponds to the lower diagram. In each diagram, the yellow square is the square the robot starts in, while the green square is the square the robot finishes in. 

**PROBLEM 3.**

There exist some cities that are built along a straight road. The cities are numbered 1,2,3,…1,2,3,… from left to right.

There are NN GBuses that operate along this road. For each GBus, the range of cities that it serves is provided: the ii-th gBus serves the cities with numbers between AiAi and BiBi, inclusive.

We are interested in a particular subset of PP cities. For each of those cities, we need to find out how many GBuses serve that particular city.

Input

The first line of the input gives the number of test cases, TT. Then, TT test cases follow; each case is separated from the next by one blank line. (Notice that this is unusual for Kickstart data sets.)

In each test case:

* The first line contains one integer NN: the number of GBuses.
* The second line contains 2NN integers representing the ranges of cities that the buses serve, in the form A1A1 B1B1 A2A2 B2B2 A3A3 B3B3 ... ANAN BNBN. That is, the first GBus serves the cities numbered from A1A1 to B1B1 (inclusive), the second GBus serves the cities numbered from A2A2 to B2B2 (inclusive), and so on.
* The third line contains one integer PP: the number of cities we are interested in, as described above. (Note that this is not necessarily the same as the total number of cities in the problem, which is not given.)
* Finally, there are PP more lines; the ii-th of these contains the number CiCi of a city we are interested in.

Output

For each test case, output one line containing Case #xx: yy, where xx is the number of the test case (starting from 1), and yy is a list of PP integers, in which the ii-th integer is the number of GBuses that serve city CiCi.

Limits

Memory limit: 1 GB.  
1≤T≤101≤T≤10.

Test Set 1

Time limit: 60 seconds.  
1≤N≤501≤N≤50.  
1≤Ai≤5001≤Ai≤500, for all ii.  
1≤Bi≤5001≤Bi≤500, for all ii.  
1≤Ci≤5001≤Ci≤500, for all ii.  
1≤P≤501≤P≤50.

Test Set 2

Time limit: 120 seconds.  
1≤N≤5001≤N≤500.  
1≤Ai≤50001≤Ai≤5000, for all ii.  
1≤Bi≤50001≤Bi≤5000, for all ii.  
1≤Ci≤50001≤Ci≤5000, for all ii.  
1≤P≤5001≤P≤500.

Sample

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U3ObAZDQFU2NW2FKI7YHMDz9UANOKmlIYa_HJr4PhWrTYArCMNtqSzzgcYqzWwFwvckivsnA-fPLQUuEw0Q/gbus_count_sample_ts1_input.txt?dl=1)

content\_copy

2

4

15 25 30 35 45 50 10 20

2

15

25

10

10 15 5 12 40 55 1 10 25 35 45 50 20 28 27 35 15 40 4 5

3

5

10

27

Sample Output

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U3oBuK25gqCX3_MyGw7GEuNWDBSU7rVvBX6sor97Y8ptpZGbb0i0FSgbh3FcTP7gOqL1r9_5lAlYoCSRRVL_w/gbus_count_sample_ts1_output.txt?dl=1)

content\_copy

Case #1: 2 1

Case #2: 3 3 4

In Sample Case #1, there are four GBuses. The first serves cities 1515 through 2525, the second serves cities 3030 through 3535, the third serves cities 4545 through 5050, and the fourth serves cities 1010 through 2020. City 1515 is served by the first and fourth buses, so the first number in our answer list is 22. City 2525 is served by only the first bus, so the second

PROBLEM 4.

Watson and Sherlock are gym buddies.

Their gym trainer has given them three numbers, AA, BB, and NN, and has asked Watson and Sherlock to pick two different strictly positive integers ii and jj, where ii and jj are both less than or equal to NN. Watson is expected to eat exactly iAiA sprouts every day, and Sherlock is expected to eat exactly jBjB sprouts every day.

Watson and Sherlock have noticed that if the total number of sprouts eaten by them on a given day is divisible by a certain integer KK, then they get along well that day.

So, Watson and Sherlock need your help to determine how many such pairs of (i,j)(i,j) exist, where i≠ji≠j and they get along well that day. As the number of pairs can be really high, please output it modulo 109+7(1000000007)109+7(1000000007).

### Input

The first line of the input gives the number of test cases, TT. TT test cases follow. Each test case consists of one line with 4 integers AA, BB, NN and KK, as described above.

### Output

For each test case, output one line containing Case #xx: yy, where xx is the test case number (starting from 1) and yy is the required answer.

### Limits

Time limit: 60 seconds.  
Memory limit: 1 GB.  
1≤T≤1001≤T≤100.  
0≤A≤1060≤A≤106.  
0≤B≤1060≤B≤106.

#### Test Set 1

1≤K≤1041≤K≤104.  
1≤N≤1031≤N≤103.

#### Test Set 2

1≤K≤1051≤K≤105.  
1≤N≤10181≤N≤1018.

### Sample

Sample Input

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U3SHtnHox2x5vWUuXl0eQux79Kp462BXrsWvTEl0jFp2YzfEBAU_--s__bcRH4ic4Auqv5WyulES-lZL0jKaB_PdcZ-mbyO9X3eA16sz_p1y26c/sherlock_and_watson_gym_secrets_sample_ts1_input.txt?dl=1)

content\_copy

3

1 1 5 3

1 2 4 5

1 1 2 2

Sample Output

[save\_alt](https://codejam.googleapis.com/dashboard/get_file/AQj_6U3DPEUAoO9D_SV0BCNSmZbrf9rGCuffGxQJAVF8TmkmcJz2mn49sff8Gnos3Z5cfdpShvDNJGZygrHYCgTtFUBQcy9gKEzDtU2LTGFpB-7oGOOVoQ/sherlock_and_watson_gym_secrets_sample_ts1_output.txt?dl=1)

content\_copy

Case #1: 8

Case #2: 3

Case #3: 0

In Case #1, the possible pairs are (1,2)(1,2), (1,5)(1,5), (2,1)(2,1), (2,4)(2,4), (4,2)(4,2), (4,5)(4,5), (5,1)(5,1), and (5,4)(5,4).

In Case #2, the possible pairs are (1,2)(1,2), (1,3)(1,3), and (4,1)(4,1).