Round C APAC Test 2016

A. gRanks

B. gFiles

C. gGames

D. gMatrix

Questions asked 2

Submissions

gRanks

6pt	Not attempted **1263/2254 users** correct (56%)
10pt	Not attempted **923/1243 users** correct (74%)

gFiles

9pt	Not attempted **529/1189 users** correct (44%)
17pt	Not attempted **222/493 users** correct (45%)

gGames

10pt	Not attempted **85/221 users** correct (38%)
18pt	Not attempted **13/44 users** correct (30%)

gMatrix

11pt	Not attempted **826/1065 users** correct (78%)
19pt	Not attempted **157/549 users** correct (29%)

Top Scores

johngs	100
cchao	100
NAFIS	100
exprosic	100
orenguy	83
nhho	82
yaray	82
BananaTree	82
mkrjn99	82
tapasjain	82

Problem A. gRanks

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 6 points	Solve A-small
Large input 10 points	Solve A-large

Problem

There are many great athletes in the world, and it's very hard to say who is the best in the world at a particular sport, especially when different athletes have won different competitions. Here's one possible system for ranking athletes:

1. Determine the number **P** of finishing places in any competition that will be worth points for athletes, and how many points $\mathbf{S_i}$ each of those finishing places is worth. For example, for **P** = 3, one possible assignment would be 1000 points for 1st place, 500 for 2nd place, and 300 for 3rd place, and 0 for anything below that. (We assume there are no ties within competitions.)

2. Since not all competitions are equally important, assign a weight $\mathbf{W_i}$ to each one. The score gained by an athlete for a competition will be the points from step 1, modified by the weight for that competition. For example, we may decide that Olympics has a weight of 5, and, continuing with our example from above, the winner of the Olympics would receive 5 * 1000 = 5000 points.

3. Since we don't want to reward athletes simply for participating in many competitions, we count only the **M** highest scores received by an athlete across all competitions. For example, if **M** = 2 and an athlete earns scores of 1000*5, 500*1, and 300*3 in three different competitions, only the 5000 and 900 would be counted.

You are given the points per finishing place, the weights of the competitions, and the results of the competitions. Can you rank all of the athletes who appeared in the competitions? If multiple athletes have the same score, they will share the same rank and listed in alphabetical order of their names.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each test case consists of the following:

1. One line with an integer **P**, the number of top places for which points are awarded.
2. One line consists with **P** integers representing the scores $\mathbf{S_i}$ for the top places, starting with first place and continuing down the places.
3. One line with an integer **N**, the number of competitions. 4. **N** lines, each of which represents a competition. Each line begins with $\mathbf{W_i}$, the weight of this competition, and continues with the **P** names of the athletes who finished in the top **P** places. They are listed in descending order starting from first place.
5. One line with an integer **M**, the maximum number of competitions counted toward an athlete's score.

Output

For each test case, output one line containing "Case #x:", where x is the test case number (starting from 1). Then output one line for each athlete, from highest rank to lowest rank, with the format `r: name`, where `r` is the rank of the athlete and `name` is the name of the athlete. You need to rank all of the athletes that appeared in the input.

Limits

$1 \le \mathbf{T} \le 10$.
$1 \le \mathbf{S_i} \le 1000$.
$\mathbf{S_i} > \mathbf{S_{i+1}}$.
$1 \le \mathbf{W_i} \le 1000$.
Each name consists only of characters A through Z, and is at most 10 characters long.

Small dataset

$1 \le \mathbf{P} \le 10$.
$1 \le \mathbf{N} \le 10$.
$1 \le \mathbf{M} \le 10$.

Large dataset

$1 \le \mathbf{P} \le 100$.
$1 \le \mathbf{N} \le 100$.

1 ≤ **M** ≤ 100.

Sample

```
Input              Output

1                  Case #1:
2                  1: BOLT
1000 500           2: GAY
6                  3: PEIMENG
5 BOLT GAY         3: TIANBING
4 GAY BOLT         5: LARRY
1 GAY TIANBING
1 GAY PEIMENG
1 TIANBING LARRY
1 PEIMENG LARRY
2
```

In the first case, Bolt scored a total of 7000 in his two competitions. Gay would have scored a total of 8500 if all competitions were counted, but since only the top 2 competitions are counted in this case, Gay scored 6500 and ranked second. Since Peimeng and Tianbing both scored 1500, they both ranked 3rd and listed by their names. Larry is last, since he scored only 1000.

Round C APAC Test 2016

A. gRanks

B. gFiles

C. gGames

D. gMatrix

Questions asked 2

Submissions

gRanks

6pt Not attempted **1263/2254 users** correct (56%)

10pt Not attempted **923/1243 users** correct (74%)

gFiles

9pt Not attempted **529/1189 users** correct (44%)

17pt Not attempted **222/493 users** correct (45%)

gGames

10pt Not attempted **85/221 users** correct (38%)

18pt Not attempted **13/44 users** correct (30%)

gMatrix

11pt Not attempted **826/1065 users** correct (78%)

19pt Not attempted **157/549 users** correct (29%)

Top Scores

johngs	100
cchao	100
NAFIS	100
exprosic	100
orenguy	83
nhho	82
yaray	82
BananaTree	82
mkrjn99	82
tapasjain	82

Problem B. gFiles

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 9 points	Solve B-small
Large input 17 points	Solve B-large

Problem

Alien tech company G has a very old file transfer tool that is still in use today. While the tool is running, it reassures users by giving status updates on both the percentage of files transferred so far and the number of files transferred so far. The status updates during the process might look like this:

```
 20% |==>-------|     1 files transferred
100% |==========|     5 files transferred
```

But the percentage isn't precise; it is simply truncated before the decimal point (i.e. floored to the next lowest or equal 1%). That is, both 1.2% and 1.7% would be displayed as 1%.

Some users may want to know the exact total number of files, so you want to modify the tool so that the status updates look like this:

```
 20% |==>-------|     1 out of 5 files transferred
100% |==========|     5 out of 5 files transferred
```

But you've realized that it may or may not be possible to know the number of files. Given the status updates that the tool displays, either figure out how many files there are, or determine that it can't be done (i.e., there are multiple possible values for the number of files). The status updates are not guaranteed to occur at regular intervals and are not guaranteed to occur whenever a file is transferred.

Input

The first line contains **T**, the number of test cases. **T** test cases follow. Each test case consists of one line with an integer **N**, the number of status updates output by the tool, followed by **N** lines with the format P_i K_i, where P_i and K_i are integers representing the percentage and number of files transferred at some point in the process. The updates are given listed in chronological order -- that is, both the P_i values and the K_i values are nondecreasing throughout a test case.

Output

For each case, output a line starts with "Case #x: y", where x is the number of the case (starting from 1), and y is either the total number of files, or -1 if that number is ambiguous.

Limits

$1 \le \mathbf{T} \le 50$.
$1 \le \mathbf{N} \le 100$.
$0 \le P_i \le 100$

Small dataset

$0 \le K_i \le 2000$
The answer is guaranteed not to exceed 2000.

Large dataset

$0 \le K_i \le 10^{15}$
The answer is guaranteed not to exceed 10^{15}.

Sample

Input	Output
3	Case #1: 5
2	Case #2: -1
20 1	Case #3: 23

```
100 5
10
25 241
27 262
43 407
44 413
57 536
64 601
67 637
84 789
95 893
96 903
10
0 0
8 2
8 2
17 4
30 7
39 9
69 16
73 17
82 19
91 21
```

Round C APAC Test 2016

Questions asked 2

Submissions

gRanks

6pt	Not attempted **1263/2254 users** correct (56%)
10pt	Not attempted **923/1243 users** correct (74%)

gFiles

9pt	Not attempted **529/1189 users** correct (44%)
17pt	Not attempted **222/493 users** correct (45%)

gGames

10pt	Not attempted **85/221 users** correct (38%)
18pt	Not attempted **13/44 users** correct (30%)

gMatrix

11pt	Not attempted **826/1065 users** correct (78%)
19pt	Not attempted **157/549 users** correct (29%)

Top Scores

johngs	100
cchao	100
NAFIS	100
exprosic	100
orenguy	83
nhho	82
yaray	82
BananaTree	82
mkrjn99	82
tapasjain	82

Problem C. gGames

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 10 points	Solve C-small
Large input 18 points	Solve C-large

Problem

The country of elves is planning to hold an elimination tournament, and there are $2^{\mathbf{N}}$ elves who would like to take part. At the start of the tournament, they will be given unique ID numbers from 1 to $2^{\mathbf{N}}$, and the Elf President will line them up in some order.

The tournament is a series of matches between two elves, and every match has one winner and one loser (there are no ties). In the first round, the first elf in the line competes against the second elf in the line, the third elf competes against the fourth elf, and so on. After the first round, the $2^{\mathbf{N-1}}$ elves who lost leave the line, and the $2^{\mathbf{N-1}}$ elves who won remain where they are. Then, the remaining elves play the second round in the same way: the first remaining elf in the line competes against the second remaining elf in the line, the third remaining elf competes against the fourth remaining elf, and so on. After **N** rounds, there will be only one elf remaining, and that elf is the winner.

M of the elves are sensitive, which means that they will be very sad if they have to compete in matches against their friends during the games. Specifically, the ith elf will be sad if they have to compete with their friends in the first $\mathbf{K}_\mathbf{i}$ rounds. (Note that friendship is not necessarily mutual: if one elf considers another elf to be a friend, the other elf does not necessarily consider that elf to be a friend.)

The Elf President wants to know: is there a way to specify the initial positions of all $2^{\mathbf{N}}$ elves to guarantee that no elf will be sad, no matter what happens in the tournament?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of one line with two integers **N** and **M**, then **M** sets of two lines each, in which the first line has integers $\mathbf{E}_\mathbf{i}$, $\mathbf{K}_\mathbf{i}$, and $\mathbf{B}_\mathbf{i}$ for one elf, and the second has $\mathbf{B}_\mathbf{i}$ integer ID numbers of that elf's friends.

Output

For each test case, output one line containing "Case #x: ", where x is the case number (starting from 1), followed by YES or NO.

Limits

$1 \le \mathbf{T} \le 200.$
$0 \le \mathbf{M} \le 2^{\mathbf{N}}.$
$1 \le \mathbf{E}_\mathbf{i} \le 2^{\mathbf{N}}.$
$1 \le \mathbf{K}_\mathbf{i} \le \mathbf{N}.$
$\mathbf{M} \le \mathrm{sum}(\mathbf{B}_\mathbf{i}) \le \min(2 * \mathbf{M}, 2^{\mathbf{N}}).$

Small dataset

$1 \le \mathbf{N} \le 3.$

Large dataset

$\mathbf{N} = 4.$

Sample

```
Input        Output

3            Case #1: NO
1 1          Case #2: YES
1 1 1        Case #3: YES
2
2 2
1 1 1
2

3 1 1
4
```

```
3 3
1 2 2
3 4
2 2 2
5 6
7 1 1
8
```

Submissions

gRanks

6pt	Not attempted **1263/2254 users** correct (56%)
10pt	Not attempted **923/1243 users** correct (74%)

gFiles

9pt	Not attempted **529/1189 users** correct (44%)
17pt	Not attempted **222/493 users** correct (45%)

gGames

10pt	Not attempted **85/221 users** correct (38%)
18pt	Not attempted **13/44 users** correct (30%)

gMatrix

11pt	Not attempted **826/1065 users** correct (78%)
19pt	Not attempted **157/549 users** correct (29%)

Top Scores

johngs	100
cchao	100
NAFIS	100
exprosic	100
orenguy	83
nhho	82
yaray	82
BananaTree	82
mkrjn99	82
tapasjain	82

Problem D. gMatrix

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 D-small
Large input 19 points	Solve D-large

Problem

You have a square **N** by **N** matrix M of nonnegative integers. We would like to make a list of the maximum values in every sub-matrix of size **K** by **K** within M, and then find the sum of those values together. (Note that the same entry of M might be the maximum value in more than one sub-matrix, in which case it will show up multiple times in the list.) Can you find that sum?

To simplify the input of the matrix, you are given two arrays **A** and **B** of length **N**, and two integers **C** and **X**. Then the entry M_{ij} (for the ith row and jth column of the matrix) equals ($\mathbf{A_i}$*i+$\mathbf{B_j}$*j + **C**) mod **X**, where i and j are in the range [1, **N**].

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case begins with one line with four integers, **N**, **K**, **C** and **X**. Then there are two lines with **N** integers each, representing the arrays **A** and **B**.

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 the maximum values in all sub-matrices of size **K** by **K**.

Limits

1 ≤ **T** ≤ 100.
1 ≤ $\mathbf{A_i}$, $\mathbf{B_i}$ ≤ 100000.
1 ≤ **C** ≤ 100000.
1 ≤ **X** ≤ 1000000007.
1 ≤ **K** ≤ **N**.

Small dataset

1 ≤ **N** ≤ 50.

Large dataset

1 ≤ **N** ≤ 3000.

Sample

```
Input          Output

3              Case #1: 3
1 1 1 5        Case #2: 19
1              Case #3: 80
1
2 1 5 11
1 2
3 4
3 2 3 109
6 4 3
2 1 5
```

In the first test case, the matrix is:
3
So the sum of maximum values is 3.

In the second test case, the matrix is:
9 3
1 6
So the sum of maximum values is 19.

In the third test case, the matrix is:
11 11 24

```
13  13  26
14  14  27
```