Algorithms for Programming Contests WS20/21 - Week 08

Chair for Foundations of Software Reliability and Theoretical Computer Science, TU München

Mikhail Raskin, Martin Helfrich, Maxi Weininger, Christoph Welzel

Welcome to our practical course! This problem set is due by

Wednesday, 13.01.2021, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E	F*	G*	H*
Difficulty	easy	easy	medium	medium	hard	easy	medium	hard
Points	4	4	6	6	8	4*	6*	8*

* catchup problem / bonus points, do not count towards total

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)

- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

Problem A
Commander-in-Chief

A rainy Sunday afternoon. What should Lea and her friends do? They wanted to go tightrope walking between two mountain summits, but this is not going to happen with rain pouring down on them, and making the rope slippery and all. So, they decide to stay in and dust off an old warfare strategy board game they played a lot when they were younger. Epic battles between huge armies, invading continents, conquering new worlds. Though, all of this is only possible for the best players. And Lea really does want to be one of those players. Roughly explained, the game is about stationing armies on different regions on a map, and letting them fight against opposing armies from a neighbouring region in order to try to conquer that region.

As far as she can remember, the best strategy for her was to subdivide each army into smaller squadrons and letting each of them fight on a different battleground. However, the real strategic trick in this game was that each of these squadrons should be of the same size, even if they belong to different armies. To maximise Lea's probability to conquer the world, the squadrons should be as big as possible, and no troops should be left behind without a squadron. Luckily, Lea has time to calculate this squadron size during the other players' turns.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing an integer n, the number of armies Lea commands. The next line contains n space-separated integers $a_1 \dots a_n$ denoting the sizes of the n armies.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the biggest possible squadron size as described above. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 500$
- $2 \leq n \leq 20$
- $1 \leq a_i \leq 10^9$ for all $1 \leq i \leq n$

Sample Input 1	Sample Output 1
2 3 12 6 21 4 24 7 12 18	Case #1: 3 Case #2: 1

Sample Input 2

```
6
5
35 19 37 11 8

2
48 48

3
44 41 26

3
33 44 33

5
9 18 45 36 36

3
29 29 29
```

Sample Output 2

```
Case #1: 1
Case #2: 48
Case #3: 1
Case #4: 11
Case #5: 9
Case #6: 29
```

Problem B
Break In

In Lea's hometown, there is a building of a branch of the Innovative Consumer Products Company (ICPC). The ICPC develops many interesting products, but is very secretive about them until they are released. Lea would like to know what their next product is, so one night, she approaches the building and examines it. The building is a large black structure without any windows and only a single door. Next to the door, there are two numeric keypads, labelled with **X** and **Y**, which are connected to a digital display. Instinctively, Lea assumes that this is the key to opening the door. She tries entering several random numbers on each keypad, but the door does not open. However, she makes some interesting observations.

Firstly, after entering numbers on the keypads **X** and **Y**, the product of the two numbers is displayed on the display above. Secondly, as the display has a fixed number of digits, if the product of the numbers is too large, the remaining digits at the front are cut off. If the product is small, the remaining digits show a `0`, i.e. the number is padded with leading zeros. For example, if the display has 3 digits, and Lea enters 3 and 5, then `015` is displayed. If Lea enters 59 and 37, then `183` is displayed.

Being unsuccessful in opening the door, Lea decides to come back the following nights for further observations of the building, while hiding in nearby bushes. At certain times in the night, she notices employees of the ICPC, always in pairs, enter the building. They do this by each one of them entering a number on each keypad, respectively.

Lea cannot see the numbers entered on the keypads, however she can view the display. The number displayed there is always a 1, with leading zeros, when the door opens (e.g. `00001`). Lea is sure that the correct numbers for opening the door have to yield this result. Further, by listening to the beeps made when the numbers are entered, she knows that the number of digits entered on either keypad is at least one and never exceeds the number of digits displayed. Finally, after the persons left, through careful analysis of fingerprints and dust build up, she concluded that the last digit entered on the keypad **Y** is 1, 3, 7 or 9. The person entering numbers on keypad **X** always wears gloves, so she could not obtain any information about the number entered there.

On one lucky night, one of the two persons was not careful enough, and Lea managed to see which number was entered on keypad **Y**. Can you tell her which corresponding number needs to be entered on keypad **X** so she can open the door?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line consisting of two integers n, the number of digits on the display, and y, the number entered on the keypad **Y**.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a number with $1 \leq x \leq 10^n - 1$ such that the last n digits of the product $x \cdot y$, possibly padded with leading 0s, are a series of $n-1$ 0s followed by a single 1. You may assume that such an x always exists. If more than one such x exists, you may output any one of them.

Constraints

- $1 \leq t \leq 1000$
- $1 \leq n \leq 18$
- $1 \leq y \leq 10^n - 1$
- The last digit of y is 1, 3, 7 or 9.

Sample Input 1

```
11
1 1
1 7
1 9
2 17
2 23
2 83
2 11
3 1
3 17
3 713
3 373
```

Sample Output 1

```
Case #1: 1
Case #2: 3
Case #3: 9
Case #4: 53
Case #5: 87
Case #6: 47
Case #7: 91
Case #8: 1
Case #9: 353
Case #10: 777
Case #11: 437
```

Sample Input 2

```
12
3 969
3 589
3 923
4 333
4 7289
4 61
4 7541
4 9231
4 613
5 60543
5 11327
5 21061
```

Sample Output 2

```
Case #1: 129
Case #2: 309
Case #3: 987
Case #4: 6997
Case #5: 9609
Case #6: 7541
Case #7: 61
Case #8: 6671
Case #9: 5677
Case #10: 13407
Case #11: 23263
Case #12: 46541
```

Problem C
N-athlon

Lea is a very active person. Whenever she is bored, she finds something new to do. This time, she invited all her friends over for a grand tournament. She devises many exciting (or sometimes just silly but fun to watch) team games, in which her friends have to compete and whoever is part of a winning team gets a point. Of course, most games need a different number of players on each team, so for every game, new teams are formed.

Nevertheless, the event is a big success and many of Lea's friends compete in suspense-packed matches of real competition sports like Soccer, Volleyball, First Person Shooters, Real Time Strategy, Hockey and some newly invented events like "Extreme Spaghetti Knitting" or "Plants vs Zombies - An Adaption" (where the contestants roleplay on Lea's front lawn and the audience votes on whoever gave the best impression).

In the evening, Lea is really tired but happy. But before she is able to sleep, she wants to determine who won the grand tournament. But in all the chaos, she realizes she doesn't even have any idea of how many people participated.

At least, she recalls the following facts: For every event she organized, she divided all the participants into teams of equal size. Since she is a math enthusiast, she thought it would be a cool idea that every team size would be a prime number, but not many people even noticed. And since she oversaw the building of those teams, for each event she thinks she remembers the amount of people that were left over and had to watch from the sidelines. She also knows how many people she invited and that almost everyone she invited was there. Can you tell her how many people participated? (Please also tell her if her memories are inconsistent and no number of friends satisfy her memorized numbers)

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with a line containing two integers n, k, where n is the number of different games Lea organized and k is the number of friends that were invited. n lines follow, each consisting of two integers $size_i$ $rest_i$, where $size_i$ is the team size for game i and $rest_i$ is the amount of people that remained after dividing the amount of participants into teams of size $size_i$.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x being the people that participated in Lea's contest satisfying the constraints on team sizes and being the largest such number less or equal to k or "impossible" if Lea has made a counting error and there is no x that satisfies these constraints. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $0 \leq n \leq 15$
- $0 \leq k < 10^{12}$
- $0 \leq rest_i < size_i < 10^8$
- $size_i$ is prime for all i
- The number of total participants x does not change between games
- $\prod^{i} size_i < 10^{18}$

Sample Input 1

```
3
2 20
3 1
5 2

1 5
2 1

2 20
11 1
17 2
```

Sample Output 1

```
Case #1: 7
Case #2: 5
Case #3: impossible
```

Sample Input 2

```
6
3 29
19 5
43 8
47 13

1 29
43 22

1 20
11 9

2 13
5 1
3 1

4 43499412
11 2
73 53
17 12
79 32

3 10497640
29 24
17 11
37 35
```

Sample Output 2

```
Case #1: impossible
Case #2: 22
Case #3: 20
Case #4: 1
Case #5: 43457537
Case #6: 10492717
```

Problem D
Soft Skills

Lea wants to get a better job and is therefore looking for some soft skill courses as she knows that every recruiter loves to read the courses' inspiring titles on a CV. She found the list of courses at IAH, the Institute for Amazing Headlines, which will look great on her CV. The people at IAH also award certificates if the participants take enough of their courses. There is not much information about the contents of the courses, but this is not what Lea is looking for, anyway.

The number of courses at IAH is overwhelming, so Lea begins to wonder how many possibilities there are to combine them for the certificate. For the certificate, only the number of courses matters. It is not important whether Lea participates in "Handling Extremely Complex Systems" or "How to Get Promoted During the Coffee Break", for instance. As the number of possible combinations for the certificate may be very big, Lea is already satisfied to find the number modulo $223092870 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$.

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line containing two integers n, the number of courses available at IAH, and m, the number of courses needed for the certificate.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the number of possibilities to pick exactly m courses modulo 223092870. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10^7$
- $1 \leq m \leq n$

Sample Input 1

```
3
5 2
7 4
100 12
```

Sample Output 1

```
Case #1: 10
Case #2: 35
Case #3: 96633810
```

Sample Input 2

```
5
7 6
11 6
6 5
10 4
11 8
```

Sample Output 2

```
Case #1: 7
Case #2: 462
Case #3: 6
Case #4: 210
Case #5: 165
```

This page is intentionally left blank.

Problem E
Keyboards

In her rare free time, Lea sometimes builds mechanical keyboards for her computer. Today she wants to experiment with a new fancy type of mechanical switches. She has two layout ideas to try out and compare. To avoid losing any switches during reassembly it would be really nice if the key count could be the same for both! But is it feasible?

For each layout there are some keys in special positions (for example, the spacebar or shift or control), and then there is a repeating part with some number of keys per column that can be stretched as long as desired. To make sure she gets enough exercise even when she has to work on an article all day long, Lea embraces the idea of the keyboards being long enough to require a bit of jogging.

Or maybe Lea wants an ortholinear keyboard, with 7 special-case keys in the middle and repeatable columns of four keys each?

Given the numbers of the special keys and the sizes of repeatable columns, can you say whether the project is feasible?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case is a line containing five integers s_1, c_1, s_2, c_2, n. Here s_1 and c_1 are the numbers of special keys and the number of keys per column in the first design, s_2 and c_2 are the corresponding numbers for the second design, and n is the minimal acceptable number of keys.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x being the minimum number of keys that is more than n and can be used to implement both designs. If there is no such x, output "impossible". Each line of the output should end with a line break.

Note that if there are enough special keys, it might be possible to implement a design using zero columns.

Constraints

- $1 \leq t \leq 20$
- $0 \leq n \leq 10^9$
- $0 \leq s_1, s_2 \leq 10^7$
- $1 \leq c_1, c_2 \leq 10^7$

Sample Input 1

```
5
10 7 8 5 40
0 6 0 10 20
0 2 1 2 0
1 998 1 997 900
0 2 0 3 999
```

Sample Output 1

```
Case #1: 73
Case #2: 30
Case #3: impossible
Case #4: 995007
Case #5: 1002
```

Problem F
Candies

Most people are quite happy to invite friends. Lea is too and of course she strives to make the invitees as happy as possible. This sometimes proves to be quite difficult. This time she plans to buy candies for everyone. While this sounds like a simple task, the eating habits of her friends complicate things.

When eating candies, all friends sit in a big circle. All Candies are poured into a big bowl that is passed around. Each friend has a specific number of candies that he eats every time he gets the bowl. The bowl starts full at Lea, is passed around each time in the same order, and Lea always eats exactly one candy whenever she gets the bowl back.

She now wants to buy a number of candies such that, no matter which of her friends show up, the bowl will end up empty after Lea takes a candy (Then the bowl is passed around no more). It may be passed around a couple of times, but it should not happen that a friend cannot take his number of candies or that it returns to Lea empty.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of Lea's friends. The next line contains a space separated list of n integers $c_1 ... c_n$, c_i is the number of candies her i-th friend eats each time he gets the bowl.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimal number of candies Lea has to buy to satisfy the constraints above. You can assume that at least one friend will show up. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 15$
- $1 \leq c_i \leq 10$
- The number x will be at most $2^{63} - 1$

Sample Input 1	Sample Output 1
2 2 3 5 3 1 1 1	Case #1: 36 Case #2: 12

Sample Input 2

```
10
1
4

3
4 2 4

2
3 2

3
3 1 4

3
2 1 4

2
3 3

2
3 2

3
1 2 1

3
2 2 3

3
1 2 2
```

Sample Output 2

```
Case #1: 5
Case #2: 3465
Case #3: 12
Case #4: 360
Case #5: 840
Case #6: 28
Case #7: 12
Case #8: 60
Case #9: 120
Case #10: 60
```

Problem G
Fantasy Chess

Lea loves chess. After having mastered the classical game completely, she turned her attention to fantasy chess, with huge boards and special pieces. Currently she studies the strategy of using the knight-like figures, that can make a hop of a fields in one direction and b fields in an orthogonal direction. An important part is of course finding a way to reach a field with coordinates larger by (x, y) from the original field. Can you write a program for that?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case is a line containing four integers a, b, x, y. Here a and b specify the move of the piece, and x and y specify the target field.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. If there is a possibility to reach the target cell, output up to eight more lines, containing three integers a', b', k each. a' and b' should be a and b in some order and possibly negated. Each such line says that the piece should make k moves (a', b'). The number k must be smaller than 10^9, every solvable test case will have a solution with all k smaller than 10^8. Together the printed moves should move the piece by x and y in the middle of a large enough board. If there are multiple solutions, any small enough one will be accepted. If the piece cannot reach the target, output the line "impossible". End the output for the case with a blank line. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $0 \leq a, b, x, y \leq 10^8$

Sample Input 1	Sample Output 1
2 1 2 1 0 2 2 1 1	Case #1: 1 2 1 1 -2 0 2 1 -1 2 -1 0 -2 1 -1 -1 2 0 Case #2: impossible

This page is intentionally left blank.

Problem H
Vaults & Vampires

Grunkh, the brutal troll, defeated the good human mage Gregor McHexroy in a long and exhausting battle. Swords and clubs went into splinters, a forest burned down and even the mountain where all squirrels from the forest ran for shelter exploded during the epic battle. Nevertheless, Grunkh survived albeit badly injured. He collects all the gold McHexroy had in his pockets and trudges back to his cave to heal his wounds. Suddenly, a wild rat appears and dares to attack Grunkh, who is 10 times as big and 100 times as strong as the rat. Normally, this would be an easy fight, but now Grunk is heavily injured and can barely move.

"I fought more than one hour to defeat this mage and now a rat tries to kill me and get all the loot? This is ridiculous, I need to find a new GM (game master)..." Lea thinks, who is playing Grunkh at the latest gathering of her friends testing the new RPG "Vaults & Vampires". Nevertheless, she has to roll the dice now and see whether she can beat this tiny rat. At least, she wants to know the exact probability to win before she does so. Can you help her?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a line containing an integer n and a string x. n is the least number of points Lea has to get when rolling the dice and x is a string describing the dice. A set of a dice with b sides each (labelled 1 to b will be described as "adb". Multiple sets of dice may be concatenated by "+" signs.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the probability to roll at least n points. The probability should be printed as a simplified rational number in the format "numerator/-denominator". Simplified means that the numerator and denominator should not have a common divisor bigger than one and should not be negative. 0 should always be printed as "0/1".

Constraints

- $1 \leq t \leq 20$
- $0 \leq n \leq 1000$
- There will be at most 50 dice with at least 3 and at most 20 sides each.

Sample Input 1	Sample Output 1
3 3 1d6 15 1d6+2d20 75 25d6	Case #1: 2/3 Case #2: 523/600 Case #3: 1478174426405911253/1579460446107205632

Sample Input 2

```
10
212 8d12+17d18
234 6d14+15d6+6d14
427 13d17+3d10+16d17
54 5d15+8d4
310 43d8
74 9d17
77 31d5
132 15d16
494 6d18+28d14
196 43d5
```

Sample Output 2

```
Case #1: 626994794977148929269606485 53/11748774497030625645561485 7216
```