Problem A: Book Borders

A book is being typeset using a fixed width font and a simple greedy algorithm to fill each line. The book contents is just a sequence of words, where each word contains one or more characters.

Before typesetting, we choose a *maximum line length* and denote this value with m. Each line can be at most m characters long including the space characters between the words. The typesetting algorithm simply processes words one by one and prints each word with exactly one space character between two consecutive words on the same line. If printing the word on the current line would exceed the maximum line length m, a new line is started instead.

```
|its.a.long...|        |its.a.long.way|
|way.to.the...|        |to.the.top.if.|
|top.if.you...|        |you.wanna.rock|
|wanna.rock.n.|        |n.roll........|
|roll.........|
```

Text from the example input with maximum line lengths 13 and 14

You are given a text to be typeset and are experimenting with different values of the maximum line length m. For a fixed m, the *leading sentence* is a sentence (a sequence of words separated with a single space character) formed by the first words of lines top to bottom. In the example above, when the sample text is typeset with the maximum line length 14, the leading sentence is "`its to you n`".

Given a text and two integers a and b, find the length of the leading sentence for every candidate maximum line length between a and b inclusive. The length of a sentence is the total number of characters it contains including the space characters.

Input

The first line contains the text to be typeset – a sequence of words separated by exactly one space character. Each word is a string consisting of one or more lowercase letters from the English alphabet.

The second line contains two integers a and b – the edges of the interval we are interested in, as described above.

It is guaranteed that $1 \leq w \leq a \leq b \leq z \leq 500\,000$, where w is the length of the longest word in the text and z is the total number of characters in the text including the space characters.

Output

Output $b - a + 1$ lines – the k-th of those lines should contain a single integer – the total length of the leading sentence when the maximum line length is equal to $a - 1 + k$.

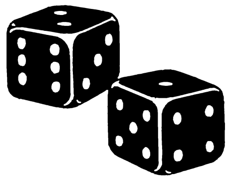
Example

input	output

`its a long way to the top if you wanna rock n roll`
`13 16`
	`22`
	`12`
	`12`
	`15`

Problem B

Butterfly Effect



There are several events that are about to happen. Each event has either a positive outcome or a negative outcome, and these outcomes affect the probabilities of the outcomes of subsequent events.

The events occur in the order given in the input. For every event i, there is an associated integer-valued *base value*, which we denote by b_i. To decide the outcome of an event, we roll a fair m-sided die with sides marked 1 through m and add the amount shown on the die to the base value. If the result is strictly positive, then the outcome is *positive*; otherwise (including if the result is zero), the *negative* outcome occurs. If the positive outcome occurs, then we modify the base values of all subsequent events according to a list of modifiers associated with the event. That is, if the outcome of event i is positive, the new base value for event j is $b_j + p_{ij}$, where p_{ij} is the modifier to event j in the case of a positive outcome for event i. If the negative outcome occurs, we do the same but with a different list of modifiers; the base value for event j becomes $b_j + q_{ij}$, where q_{ij} is the associated modifier.

You have the power to intervene in a certain number of events. When you intervene, instead of rolling one die, you roll two dice and then choose the die you prefer. For each event, you decide whether or not to intervene immediately before that event's die is rolled, i.e., you may use the outcomes of previous events to decide whether or not to intervene. Can you maximize the probability of the final event having a positive outcome?

Input

The first line contains three space-separated integers n, k, and m ($1 \leq k \leq n \leq 20$, $4 \leq m \leq 1{,}000$), denoting the number of events, the maximum number of interventions, and the die size, respectively. Next are $3n$ lines describing the base values and modifiers of the events, in the following format:

- Line $3i - 1$: One integer b_i denoting the base value of event i. The base value of each event will have absolute value at most 2,000.
- Line $3i$: $n - i$ space-separated integers $p_{i,i+1}, \ldots, p_{in}$ denoting the modifiers to the base values of events $i + 1$ through n in the case of a positive outcome for event i. Each modifier will have absolute value at most 2,000.
- Line $3i + 1$: $n - i$ space-separated integers $q_{i,i+1}, \ldots, q_{in}$ denoting the modifiers to the base values of events $i + 1$ through n in the case of a negative outcome for event i. Each modifier will have absolute value at most 2,000.

The final event has no modifiers, and thus the last two lines of the input are empty.

Output

Print, on a single line, a single number equal to the maximum probability of the final event having a positive outcome, rounded and displayed to exactly 6 decimal places.

Sample Input	Sample Output
2 2 6 -3 -100 100 0	0.750000

Sample Input	Sample Output
4 1 10 -5 -10 9 0 9 -10 0 -10 0 10 0 0 -10 10 0 -10	0.990000

Problem C: Bottled Up

Peter is expecting a large shipment of fuel oil, but he has a small problem (doesn't everyone in these programming problems!). The only containers he has are a set of large bottles (each with the same volume) and a set of smaller bottles (also each with the same, but smaller volume). Given the volume of the shipment of oil, he would like to store the oil in the bottles so that:

1. all of the oil is stored,
2. each bottle is filled to the top, and
3. the minimum number of bottles is used.

While Peter thinks he has solved this problem for his given bottle sizes, he often spends hours wondering what would happen if his bottles had different volumes (apparently Peter doesn't lead the most exciting life).

Input

The input consists of a single line containing three positive integers, s, v_1 and v_2, where $s \leq 1\ 000\ 000$ is the volume of the shipment, and $v_1, v_2 \leq 1\ 000\ 000$ are the volumes of the two types of bottles, with $v_1 > v_2$.

Output

Output the number of bottles of size v_1 and the number of bottles of size v_2 which satisfy Peter's three conditions. If the conditions cannot be met, output `Impossible`.

Sample Input and Output

Sample Input 1	Output for Sample Input
`1000 9 7`	`108 4`

Sample Input 2	Output for Sample Input
`1000 900 7`	`Impossible`

Sample Input 3	Output for Sample Input
`1000 10 7`	`100 0`

D: Desert Bit Map

This problem requires you to search a black and white satellite image of a desert for a secret building complex with a given shape. A complex of this given shape may host an installation for producing the strategic xeenium macgillicudamate ingredient, and must keep its orientation with regard to cardinal axes (North-East-South-West). Rotations and mirror images are not allowed because they would interfere with the delicate alchemy required for the production process. You must determine how many times the given complex may possibly occur in the image.

Consider the following images, both on the same scale, where a # (sharp) is a "black" pixel representing a part of a building, and a . (dot) is a "white" pixel, representing sand. On the left is an image of the complex you are trying to locate, on the right is an image of the desert with some buildings on it.

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

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

- How many possible locations for the given secret buildings do we count?
- The answer is *four*: one at the top-left corner, two overlapped possibilities to its right, and one in the bottom right. The shapes near the top-right corner, and in the centre bottom don't count because they are rotated (remember that rotated and/or mirrored images do not count).
- Note that, as this answer implies, the sand pixels in the image of the building complex simply establish the necessary relationships between the building parts. In the actual image they may contain *either* sand *or* other building parts (possibly for disguising the true nature of the complex).
- *Assume* that images representing strategic complexes are already trimmed of any unneeded dot "white" pixels on the edges, i.e., these images will always contain *at least one # character on each edge* (as our example shows). An edge here is the first or last row or column.

Input

Each test case will give you the specification for the building complex image followed by the specification for the desert image.

In each problem the input is:

- Line 1: 2 positive integers, b_1, b_2, respectively representing the number of lines and the number of columns in the following buildings image. Both numbers will be in the range 1 to 16 inclusive.
- Next b_1 lines: b_2 characters (# or .) on each line to represent part of the image of the building complex.
- Next Line: 2 positive integers, d_1, d_2, respectively representing the number of lines and the number of columns in the following Desert image. Both numbers will be in the range 1 to 64 inclusive.
- Next d_1 lines: d_2 characters (# or .) on each line to represent the desert image.

Output

The output for each test case consists of a single integer value on a line by itself being the number of matches found.

Sample Input and Output

Sample Input 1	Output for Sample Input
2 2 #. ## 3 5 #.#.# ##### .###.	4

Sample Input 2	Output for Sample Input
1 3 #.# 3 6 ##..## .#.#.# #.#...	3

Sample Input 3	Output for Sample Input
3 3 #.. #.# #.. 5 36 #......#........................#... #.#....#.#...#.........#............ #......#.#...##....#...##......###.. ....#.....#...#.#...##........#.# ....#.....#.......#..	4

Problem E: Jelly

A local school provides jelly for their pupils every day and the school staff are very careful to see that each child has exactly the same amount.

The jelly is prepared the previous day by pouring the liquid jelly into rectangular sided moulds, one mould per child, and then put in the fridge where it sets. The moulds may differ by the length and width of their sides but are filled to different heights so that they all have the same volume. Length, width, and height are always integer values.

Unfortunately, one of the cleaners loves practical jokes! Whenever he can, before the jelly has set, he tips liquid jelly from one of the moulds into another. He is happy if he succeeds just once and doesn't repeat the joke with other moulds.

Your task is to help the school staff by preparing a report for them. They need to know who has lost jelly and who has gained it so that they can correct matters before the children arrive.

Input

The input consists of one test case. The test case begins with a single integer n, $1 \leq n \leq 100$, representing the number of children for whom jelly was prepared. Following this are n lines, each line representing one child. The data for a child consists of the child's name followed by a single space and three space-separated integers, l, w and h ($1 \leq l, w, h \leq 100$) being the length, width and height of the jelly in that child's mould. A child's name consists of a sequence of 1 up to 10 alphabetic characters (upper and/or lower case). No two children have the same name.

Output

Output consists of one line of text. If the cleaner did not manage to transfer any jelly before it set, output:

`No child has lost jelly.`

If the cleaner did manage to transfer jelly, your output must be in the form:

`*ChildA* has lost jelly to *ChildB*.`

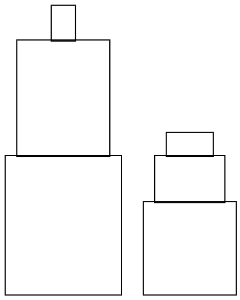
Sample Input and Output

Sample Input 1	Output for Sample Input
3 Joe 10 10 2 Susan 10 5 4 Bill 5 5 8	No child has lost jelly.

Sample Input 2	Output for Sample Input
4 Zoe 10 2 2 Lee 6 5 2 Alan 5 4 4 Tommy 12 5 1	Zoe has lost jelly to Alan.

Problem F: Towering Problem

You've been put in charge of an art exhibit from the famous minimalist sculptor J (even his name is minimalist!). J's work involves the careful layout of vertically dispositioned orthogonal parallelepipeds in a set of tapering obelisks — in other words, he puts smaller boxes on top of larger boxes. His most recent triumph is called "2 by 3's Decreasing," in which he has various sets of six boxes arranged in two stacks of three boxes each. One such set is shown below:



J has sent you the art exhibit and it is your job to set up each of the six-box sets at various locations throughout the museum. But when the sculptures arrived at the museum, uncultured barbarians (i.e., delivery men) simply dropped each set of six boxes on the floor, not realizing the aesthetic appeal of their original layout. You need to reconstruct each set of two towers, but you have no idea which box goes on top of the other! All you know is the following: for each set of six, you have the heights of the two towers, and you know that in any tower the largest height box is always on the bottom and the smallest height box is on the top. Armed with this information, you hope to be able to figure out which boxes go together before tomorrow night's grand opening gala.

Input

The input consists of eight positive integers. The first six represent the heights of the six boxes. These values will be given in no particular order and no two will be equal.

The last two values (which will never be the same) are the heights of the two towers.

All box heights will be ≤ 100 and the sum of the box heights will equal the sum of the tower heights.

Output

Output the heights of the three boxes in the first tower (i.e., the tower specified by the first tower height in the input), then the heights of the three boxes in the second tower. Each set of boxes should be output in order of decreasing height. Each test case will have a unique answer.

Sample Input and Output

Sample Input 1	Output for Sample Input
`12 8 2 4 10 3 25 14`	`12 10 3 8 4 2`

Sample Input 2	Output for Sample Input
`12 17 36 37 51 63 92 124`	`63 17 12 51 37 36`

Problem G: Arithmetic

It doesn't get much simpler than this. Calculate the sum of the two input numbers and the difference between the two input numbers.

Make sure you look at the input ranges and think about the data type needed to calculate the possible results.

Input

The input consists of two integer values, a and b ($-2\ 000\ 000\ 000 \leq a, b, \leq 2\ 000\ 000\ 000$), on a single line separated by a single space .

Output

Print, on a single line, firstly the sum of a and b and secondly the difference between a and b.

Sample Input and Output

Sample Input 1	Output for Sample Input
`10 2`	`12 8`

Sample Input 2	Output for Sample Input
`2 10`	`12 -8`

Sample Input 3	Output for Sample Input
`5 5`	`10 0`

Sample Input 4	Output for Sample Input
`-5 -5`	`-10 0`

Sample Input 5	Output for Sample Input
`2000000000 2000000000`	`4000000000 0`

Problem H: Footballers

The All Golds Rugby Club categorises members as Senior or Junior. Anyone over 17 years old is a `Senior`, as is anyone weighing 80 kg or more. Everyone else is a `Junior`. Your task is to correctly classify club members.

Input

The input consists of a single line containing a player's name followed by two positive integers, a ($5 \leq a \leq 60$) and w ($25 \leq w \leq 180$) being the age in years and weight in kilograms of the player. The player's name consists only of alphabetic characters i.e. there are no spaces in the name.

Output

Print, on a single line, the player's name, followed by a space, followed by the player's category, either `Junior` or `Senior`.

Sample Input and Output

Sample Input 1	Output for Sample Input
`Joe 16 34`	`Joe Junior`

Sample Input 2	Output for Sample Input
`Bill 18 65`	`Bill Senior`

Sample Input 3	Output for Sample Input
`Billy 17 65`	`Billy Junior`

Sample Input 4	Output for Sample Input
`Sam 17 85`	`Sam Senior`

I Highway Hassle

IF YOU SPEND NINE MINUTES OF YOUR TIME TO SAVE A DOLLAR, YOU'RE WORKING FOR LESS THAN MINIMUM WAGE.
source: xkcd.com/951

A transport company has come to you for help. One of the biggest expenses that they have is the petrol for the trucks. They would like to minimize the money they need to spend on petrol.

Because of the long trips, a truck driver typically needs to stop multiple times at a petrol station to tank up. What complicates matters is that the price of petrol is not the same at every station. The differences could in fact be so significant that it pays to take a detour in order to visit a station with a low price. Yet another complication is that the price is not the same every day (but it does not change during the day).

The good news is that they can find out, every morning, what the price of petrol is at every station for that day. They also have, for every destination, a simple graph representing the relevant part of the road network, containing only the major intersections and petrol stations as nodes. Furthermore, they know for every road exactly how much petrol is needed to go from one node to the other, down to the milliliter; it does not depend on the direction or the amount of petrol in the tank. The truck drivers also have the ability to tank with milliliter precision.

It is perfectly fine for a truck to run out of petrol at the exact moment it arrives at a petrol station or at the destination; there is in fact a spare tank to allow for small fluctuations in fuel consumption, but that petrol is not supposed to be used. You may therefore ignore its existence.

A final thing to take into consideration is that the trucks have a fuel tank of limited size. With all that information, can you work out what the optimal path to the destination is, along with the optimal tanking strategy?

Input

On the first line one positive number: the number of test cases, at most 100. After that per test case:

- one line with three space-separated integers n, m and s ($2 \leq n \leq 1\,000$ and $1 \leq m \leq 10\,000$ and $1 \leq s \leq 120$): the number of nodes, edges and petrol stations, respectively.
- one line with a single integer t ($1 \leq t \leq 100\,000$): the maximum amount of petrol that the fuel tank can hold in milliliters.
- m lines, each with three space-separated integers a, b and f ($1 \leq a, b \leq n$ and $a \neq b$ and $1 \leq f \leq 100\,000$), indicating that there is a road between nodes a and b which takes f milliliters of petrol (fuel) to traverse.
- s lines, each with two space-separated integers x and p ($1 \leq x \leq n$ and $1 \leq p \leq 100$): the node x where each petrol station is located and the price p per milliliter of petrol at that station, respectively.
- one line with two space-separated integers c and d ($1 \leq c, d \leq n$ and $c \neq d$): the nodes where the company and the destination are located, respectively.

Every road is bidirectional. There is at most one road between any pair of nodes. There is always a petrol station at node c (it is right next to the company). The truck starts with an empty tank. The destination is guaranteed to be reachable.

Output

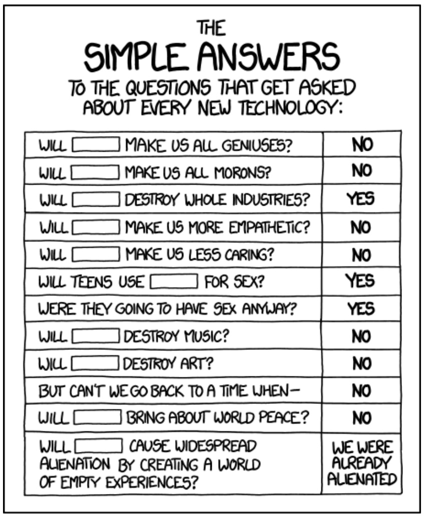
Per test case:

- one line with a single integer: the minimum amount of money that needs to be spent on petrol.

Sample in- and output

Input	Output
3 3 3 2 2000 1 3 800 1 2 500 2 3 500 1 70 2 40 1 3 5 5 3 1000 1 2 800 2 5 800 1 3 400 3 4 600 4 5 600 1 80 2 90 3 20 1 5 4 3 3 1000 1 2 200 2 3 600 3 4 300 1 40 2 70 3 90 2 4	55000 134000 61000

J Key to Knowledge



source: xkcd.com/1289

A while ago a class of students took an exam. It consisted of true-or-false questions only, but the questions were really tough! Even afterward, with the help of the book, lecture notes and each other, the students are not sure what all the correct answers were. They could ask the professor of course, but he is not the kind of guy you would dare to disturb with questions that betray your ignorance. Anyway, the students at least have an idea of how they performed.

The professor has just returned the exams, but the results are not quite as expected. Unfortunately, he did not indicate on their exams which answers were right or wrong, he has simply put the number of correct answers at the top. So what were the correct answers according to the professor then? Given how far off some of the grades are from expected, there is a suspicion among some of the students that the professor did not count the correct answers properly.

Given, for each student, the answers he/she gave and the number of correct answers, can you work out what set of correct answers matches the results?

Input

On the first line one positive number: the number of test cases, at most 100. After that per test case:

- one line with two space-separated integers n and m ($1 \leq n \leq 12$ and $1 \leq m \leq 30$): the number of students and the number of questions in the exam, respectively.
- n lines, each with m digits, each digit either `0` or `1`: the answers given by each student, with `0` representing "false" and `1` representing "true". This is followed by a space and a single integer c ($0 \leq c \leq m$): the number of correct answers for that student.

Output

Per test case:

- one line with m digits, each one either `0` or `1`: the unique set of correct answers that could account for all the results. If there is not exactly one such set, the line should read "`# solutions`" instead, with the number of solutions in place of '`#`'.

Sample in- and output

Input	Output
3 3 5 01101 4 10100 3 00011 3 3 5 01101 0 10100 3 00011 2 4 4 0000 2 1010 2 0101 2 1111 2	00101 0 solutions 4 solutions