Problem A

Hacking the Screen

The ZOO management had been wondering for a long time how to increase the number of children visitors to the ZOO. The solution was surprising and unexpected in many ways. They installed a huge screen past the entrance and started to display short quizzes on it. The child in the crowd who first shouts out the answer to the quiz question is granted one day free access to the ZOO. The screen became soon very popular and various types of quizzes are routinely shown there. One type of the quiz is the math quiz containing arithmetic operations on integers. The management worries that older siblings and friends of the children might develop a math quiz screen hacking strategy: Snap the screen with the phone, run the image recognition SW which extracts formulas from the image, evaluates them, and presents the solution to the phone holder who immediately shouts out the answer.

Your task is to assess the difficulty of producing the screen hacking software. To get a better feel of the problem you will first develop a simple toy model application. Your code will read the formula presented in the preprocessed form of ASCII art and evaluate it.

Input Specification

There are multiple test cases. First line of each test case contains two integers R and C ($1 \leq R \leq 3, 1 \leq C \leq 1\,000$). Each of the following R lines contains C characters. The whole matrix of $R \times C$ characters represents a single arithmetic formula written in ASCII art and generated by the following set of rules:

```
 FORMULA -> COMPLEX | FORMULA + COMPLEX | FORMULA - COMPLEX

 COMPLEX -> SQRT | FRACTION | TERM
              ______
    SQRT -> \/SIMPLE

            SIMPLE
FRACTION -> ======
            SIMPLE

  SIMPLE -> TERM | SIMPLE + TERM | SIMPLE - TERM

    TERM -> INTEGER | INTEGER * TERM

 INTEGER -> 0 | 1 | 2 | 3 | ... | 999999 | 1000000
```

There are also a few additional specifications regarding the layout of the formula.

- The horizontal bar of each `SQRT` is made of one or more underscore symbols ('_', ascii decimal code 95) and it always occupies the uppermost line of the formula in the screen.

- When the formula occupies exactly two lines, then the first line contains only horizontal bars of all `SQRT` parts of the formula.
- When the formula occupies exactly three lines, then all `TERM`s and all arithmetic operation symbols which are not part of any `FRACTION` or `SQRT` occupy the second line of the formula in the screen.
- The length of the horizontal bar of `SQRT` is the same as the length of `SIMPLE` under the bar.
- The fraction bar in `FRACTION` consists of one or more equality signs, its length is equal to the maximum of the lengths of `SIMPLE` above the bar and `SIMPLE` below the bar.
- There is always exactly one space preceding and following each arithmetic operation symbol (`+`, `-`, `*`) on a particular line.
- The formula exactly fits in to the $R \times C$ matrix, there are no blank/empty columns in front of the whole formula or behind it.

The whole formula is evaluated according to the standard arithmetic rules. Namely: Each `FORMULA` and each `TERM` is evaluated from left to right. Each `SIMPLE` is also evaluated from left to right with the additional standard condition that the multiplication has higher priority than the addition/subtraction. Evaluation of `SQRT` and `FRACTION` is also standard. The value of any evaluated `FORMULA`, `COMPLEX`, `SQRT`, `FRACTION`, `SIMPLE` and `TERM` is an integer whose absolute value does not exceed 1 000 000.

There is one empty line after each test case. The input is terminated by a line with two zeros.

Output Specification

For each test case print a separate line with the value V of the input formula.

Sample Input

```
1 13
1 + 2 * 3 - 4

2 16
  ---------
\/3 * 4 - 3 + 10

3 5
6 * 4
=====
  12

3 13
    22     --
3 - == - \/16
    11

0 0
```

Output for Sample Input

```
3
13
2
-3
```

Problem B

Chasing the Cheetahs

A National Geographic film crew is visiting the ZOO this week. They are creating a documentary about animal speed and they would like to film one or more cheetahs running at full pace. A solitary running cheetah has been filmed successfully many times. Therefore, the crew is also after a much more spectacular feat: As many as possible cheetahs sprinting on parallel tracks filmed together in one shot.

"No, that is impossible," said the director. "We cannot squeeze those animals into some sort of a start box, as you probably imagine, and then open the box and make them run all at once. It is clearly too dangerous and unpredictable. No."

"Then let us make more boxes and open some of them earlier and some of them later," said one of the filmmakers. "Could that work?"

"And if we open the boxes with the slower cheetahs a bit earlier then after a while the faster ones will be overtaking the slower ones and that would be a great shot," pointed out another filmmaker. "We would like to see the whole pack rather short with the last animals close the leading ones. As close as possible and at least for a moment."

It was a long and difficult discussion which ensued, but in the end the circumstances of the experiment were agreed upon.

You are given the start time and the speed of each cheetah. The length of the pack, which is defined as the distance between the first and the last cheetah in the pack, might be different at different moments. Find the minimum length of the pack during the run, where all cheetahs must be running. You may also suppose that the track is so long that the minimum length of the pack happens at least a moment before the first cheetah reaches the finish line.

All start boxes will be so close that you may consider them to be in the same place. The k-th cheetah will be released from its start box at the given time t_k. at the same distance form the finish line. The k-th cheetah is expected to run the whole distance at constant speed v_k.

Input Specification

There are more test cases. Each case occupies more lines. The first line of a case contains the number of cheetahs N ($1 \le N \le 100\,000$). Next, there are N lines, each line contains two integers t_k, v_k separated by spaces and representing the start time and velocity of the k-th cheetah ($1 \le k \le N$). All input values t_k and r_k are positive and less than 10^5. The input is terminated by a line containing zero.

Output Specification

For each test case, print a single line with one floating point number L specifying the minimum length of the running pack. Your answer should not differ from the correct answer by more than 10^-2. The length of the pack is the distance between the first and the last animal in the pack. The length can be measured at any time $T \geq \max(t_k, k = 1, ..., N)$. We suppose that each cheetah is running at a constant speed for all the time from the start and also at its moment of release from the start box.

Sample Input

```
2
1 1
1 1
2
1 99999
99999 99999
3
1 1
3 2
4 3
0
```

Output for Sample Input

```
0.000
9999700002.000
0.500
```

Problem C
Falcon Dive

"Our high speed camera failed at the most inappropriate moment," said the director of the ZOO. "This sequence with the falcon hurtling towards the ground at 250 km/h is absolutely stunning. I had hopes that we could use the last frame as a promotion picture, it would look great with the autumn trees in the background. But the falcon is too high, even in this very last frame caught by the camera before it broke."

"Cut out the falcon from the picture in Photoshop and just move it downwards," said the falconer. "It's a routine photo manipulation."

"That would be unnatural," objected the director. "We cannot show the public such obviously doctored pictures."

"On the contrary, that would be quite natural," replied the falconer. "Look, the falcon in such speed does not change its orientation so much, its shape in the picture remains virtually the same in a few consecutive frames. So if you move it down artificially it would still be a very good approximation of the natural situation which really occurred during the filming."

After some hesitation, the director agreed with the proposition.

You are given two last frames of the camera with the silhouette of the falcon in both frames. The background in the frames is identical, only the silhouette of the falcon is at a different position in both frames. The falcon is moving at a constant speed and the time between consecutive camera frames is also constant. Your task is to reconstruct the missing next frame in which the position of the falcon silhouette is changed according to its speed and to the speed of the camera. The background in the new frame should be the same as the background in the previous two frames.

Input Specification

There are more test cases. Each test case starts with a line containing two integers M, N ($2 \leq M, N \leq 1\,000$) and a printable ASCII character C enclosed in single quotes. The values on the line are separated by spaces. Next, there are M lines, one empty line, and other M lines. The first M lines represent the first frame, the last M lines represent the second frame. Each nonempty line contains string of exactly N printable ASCII characters. Each character represents one pixel of the original frame. Each frame contains a complete silhouette of the falcon. In both frames all silhouette pixels are represented by the character C and all pixels which do not belong to the silhouette are represented by characters other than C. The pixels of the silhouettes in both frames do not overlap even partially, in other words, no coordinates of a pixel of the silhouette in the first frame are the same as the coordinates of any pixel of the silhouette in the second frame. The shapes of the silhouettes in both frames are identical. The silhouette in any frame can be shifted by some number of pixels horizontally and/or vertically so that its position exactly matches the position of the silhouette in the other frame. The silhouettes do not rotate. For various technical reasons the silhouette image might not be connected, it may comprise of more disconnected regions in the frame.

A printable ASCII character is an element of the subset of ASCII characters starting with the exclamation mark character (`'!'`, ASCII code 33 in decimal) and ending with the tilde character (`'~'`, ASCII code 126 in decimal).

There is a blank line between successive cases. The input is terminated by a line containing "`0 0 ' '`".

Output Specification

For each test case, print a picture frame consisting of M lines with N characters each. The frame should represent the result of exact extrapolation of the falcon's movement based on the two input frames. If the silhouette image in the second input frame is shifted horizontally and vertically by some number of pixels relatively to the first input frame then the silhouette image in the result frame should be shifted horizontally and vertically by the same number of pixels relatively to the second frame. It is possible that the falcon's silhouette might appear in the frame picture only partially or it may not appear there at all. Print one empty line after each case.

Sample Input

```
2 2 'X'
X^
--

.X
--

3 12 'A'
ABABABABABAC
BABABABABABB
ABABABABABAB

BABABABABABA
BBABABABABAB
BABABABABABA

6 26 '>'
../||\........00......\|/.
>//||\\.....000000....-0-.
/>>>|\\\....000000..../|\.
..>||........0000.........
...||.........||..........
||||||||||||||||||||||||||

../||\.....>..00......\|/.
.//||\\.....>>>000....-0-.
///||\\\....0>0000..../|\.
...||........0000.........
...||.........||..........
||||||||||||||||||||||||||

0 0 ' '
```

Output for Sample Input

```
.^
--

BBABABABABAC
BBBABABABABA
BBABABABABAB

../||\........00......\>>>
.//||\\.....000000....-0>.
///||\\\....000000..../|\.
...||........0000.........
...||.........||..........
||||||||||||||||||||||||||
```

Problem D

The Fox and the Owl

Fox Mithra has finally learned the numbers and he is now familiar with the concept of 'one', 'two', 'three' and also even 'zero', 'minus one', 'minus two' and so on. Really, an achievement for such a small fox. He took the textbook and copied the integers from the book one by one from the smallest to the biggest on the wall of his enclosure in the ZOO.

"Look, there is something wrong with your sequence on the wall", said the owl who just landed on the branch above Mithra's head. "You should put 30 between 20 and 22, there."

"Why?"

"Because the importance of a number is judged by the sum of its digits. 30 is therefore less important than 22 and it is more important than 20. And obviously, 30 should be equally close to 20 and 22 because its sum of digits differs only by one from both 20 and 22."

"I see," replied Mithra, "you are really clever. Can you help me please to rearrange the sequence correctly? Each time I tell you a number N you will tell me the closest smaller number with sum of digits bigger by one than the sum of digits of N."

"With pleasure," nodded the owl majestically.

Your task is to imitate the owl's task. Given an integer N you have to find the biggest integer which is smaller than N with sum of digits bigger by one than the sum of digits of N.

Input Specification

There are more test cases. Each case consists of one line containing a single integer N ($|N| \leq 10^{100\,000}$). The input is terminated by a line containing string "`END`" and no other symbols.

Output Specification

For each test case print on a separate line the number demanded by fox Mithra.

Sample Input	Output for Sample Input
`30`	`22`
`199`	`-299`
`1000`	`200`
`1520`	`1512`
`END`	

Problem E

Feeding the Herrings

Zookeper Willy is feeding herrings today. He is feeding them to seals, as they are their preferred food. There are three separate pools in which the seals live. The ZOO is a modern institution and it demands their employees to keep track of feeding habits of animals. There is a touchscreen installed at the seals pools and Willy has to enter the number of herrings which he is going to deposit into each of the three pools. Unfortunately, the screen is not working properly - in particular, it is impossible to enter the digit '3'.

Willy called the chief marine mammals zookeper and asked for help.

"That is OK," said the chief, "just distribute the herrings in such a way that the number of herrings which go into each pool does not contain the digit '3'."

"But there is a lower limit L on the number of herrings which have to be put into each pool," reacted Willy, "I might not be able to find a suitable division."

"You will be able to find a suitable division," assured him the chief, "considering the numbers of herrings in the bucket, there should be zillions of possible divisions."

"Well, exactly how many?" wondered Willy for himself.

You will be given the total number N of herrings which are to be deposited into the seals pools and the lower limit L on the number of herrings in each of the pools. Find out in how many ways might these N herrings be placed into the pools in such a way that the number of herrings in each pool does not contain digit '3' in its decimal representation. In this problem, we do not distinguish between individual herrings as they are all more or less of the same size and nutrition value. We do distinguish between the pools, though, because they are populated by different groups of seals. Also, we suppose that no herring can be divided into pieces.

Input Specification

There are more test cases. Each case consists of a single line containing two integers N, L ($1 \leq N \leq 10^{10\,000}, 1 \leq L \leq N/3$) separated by space and representing the number of herrings in the bucket and the lower limit on the number of herrings which have to be deposited in each of the pools. The input is terminated by a line with two zeros.

Output Specification

For each test case print on a separate line the number of possible divisions of the herrings into the three given pools. Express the result modulo 12345647.

Sample Input

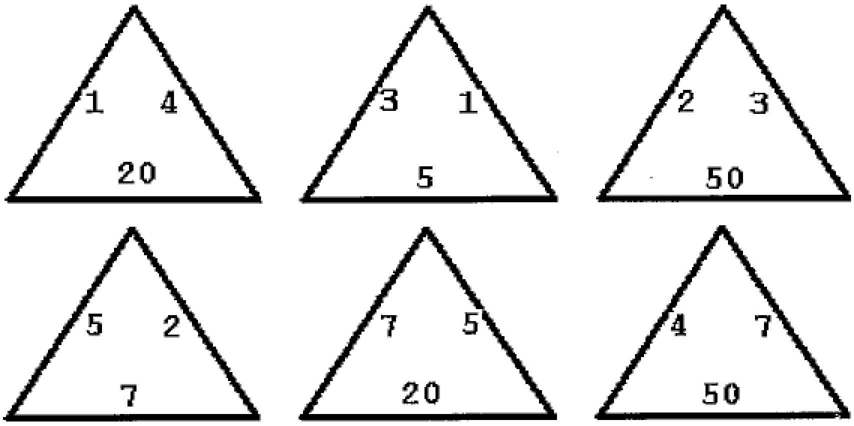
```
3 1
4 1
7 2
99999 1
0 0
```

Output for Sample Input

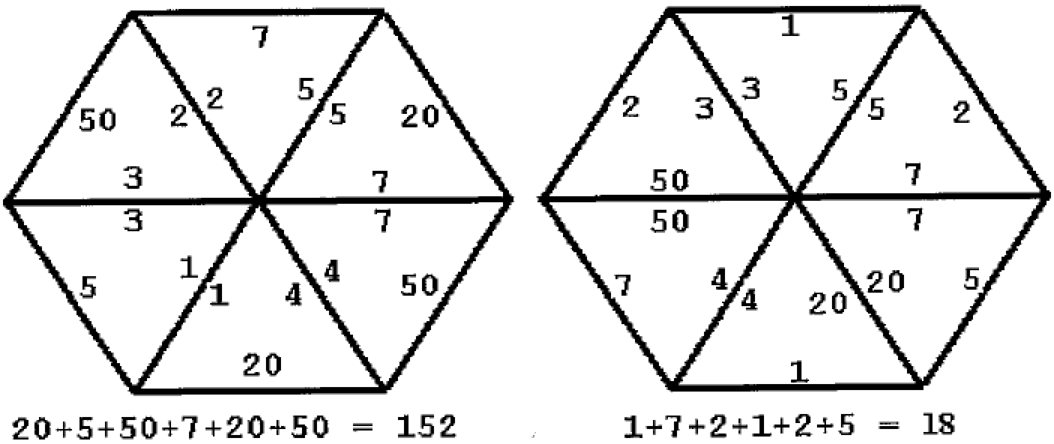
```
1
3
0
9521331
```

Problem F

Triangle Game



In the triangle game you start off with six triangles numbered on each edge, as in the example above. You can slide and rotate the triangles so they form a hexagon but the hexagon is only legal if edges common to two triangles have the same number on them. You may not flip any triangle over. Two legal hexagons formed from the six triangles are illustrated below.



The score for a legal hexagon is the sum of the numbers on the outside six edges.

Your problem is to find the highest score that can be achieved with any six particular triangles.

Input

The input will contain one test case which consists of six lines with three integers separated by spaces. All values will be between 1 and 100 inclusive. Each line contains the numbers on the triangles in clockwise order. Compare the top image with the first sample input for clarification.

Output

On a single line output only the word `none` if there are no legal hexagons or the highest score if there is a legal hexagon.

Sample Input and Output

Sample Input 1	Output for Sample Input
1 4 20 3 1 5 50 2 3 5 2 7 7 5 20 4 7 50	152

Sample Input 2	Output for Sample Input
10 1 20 20 2 30 30 3 40 40 4 50 50 5 60 60 6 10	21

Sample Input 3	Output for Sample Input
10 1 20 20 2 30 30 3 40 40 4 50 50 5 60 10 6 60	none

G • Compositions

A *composition* of an integer ***n*** is an ordered set of integers which sum to ***n***. Two *compositions* with the same elements but in different orders are considered different (this distinguishes *compositions* from *partitions*). For example, all the *compositions* of the first few integers are:

```
1: {1}
2: {1+1, 2}
3: {1+1+1, 1+2, 2+1, 3}
4: {1+1+1+1, 1+1+2, 1+2+1, 1+3, 2+1+1, 2+2, 3+1, 4}
```

Note that `1+2` and `2+1` each count as distinct compositions of `3`. As you may have suspected, there are $2^{(n-1)}$ *compositions* of ***n***.

In this problem, we set conditions on the elements of the *compositions* of ***n***. A *composition* misses a set ***S*** if no element of the composition is in the set ***S***. For example, the *compositions* of the first few integers which miss the set of even integers are:

```
1: {1}
2: {1+1}
3: {1+1+1, 3}
4: {1+1+1+1, 1+3, 3+1}
```

No odd integer can have a *composition* missing the set of odd integers and any *composition* of an even integer consisting of only even integers must be `2` times a composition of $n/2$.

For this problem you will write a program to compute the number of *compositions* of an input integer ***n*** which miss the elements of the arithmetic sequence $\{m + i*k \mid i = 0,1,\ldots\}$.

Input

The first line of input contains a single decimal integer ***P***, ($1 \le P \le 10000$), which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of a single line of input. It contains the data set number, ***K***, followed by the three space separated integers ***n***, ***m*** and ***k*** with ($1 <= n <= 30$) and ($0 <= m < k < 30$).

Output

For each data set there is one line of output. The single output line consists of the data set number, ***K***, followed by a single space followed by the number of *compositions* of ***n*** which miss the specified sequence.

Sample Input	Sample Output
3 1 10 0 2 2 15 1 4 3 28 3 7	1 55 2 235 3 18848806

H: Vowels

The English alphabet consists of 26 letters. Five of these (`a`, `e`, `i`, `o` and `u`) are classified as vowels, the remaining 21 as consonants. Almost every English word contains at least one vowel (`rhythm` is one of the few exceptions).

In this problem you will be given a piece of English text. Your task is to determine the frequency of each vowel that is found in the piece, and to display the answers sorted by frequency, highest frequency first. Where two vowels are equally frequent, they are to be displayed in alphabetical order.

As you can see from the examples below, upper case and lower case letters are considered to be the same letter in this problem. Use lower case in your output. As you can see from the second example, a frequency of zero must still be displayed.

Input

Input will consist of a single line of text with length 0 to 200 characters, inclusive.

Output

Output a single line with each vowel in lower case, followed by a colon, followed by the frequency of that vowel. There must be one space between each vowel and its count.

Sample Input and Output

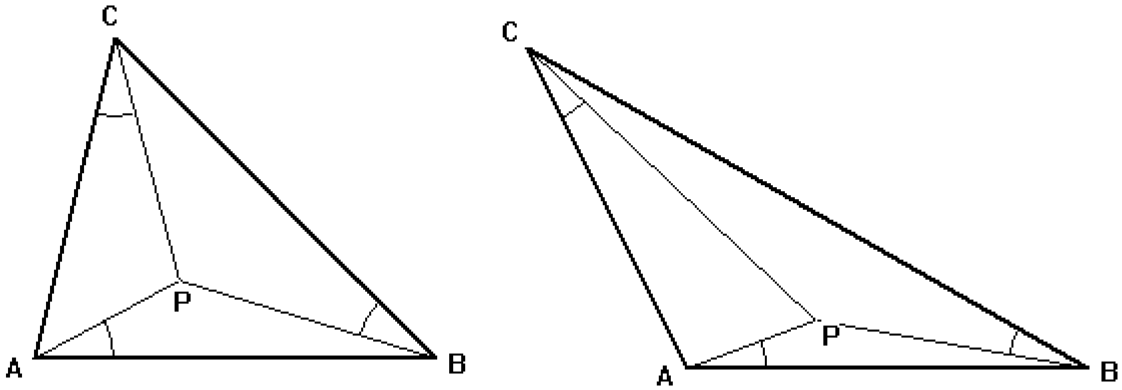
Sample Input 1	Output for Sample Input
`Ugh!!`	`u:1 a:0 e:0 i:0 o:0`

Sample Input 2	Output for Sample Input
`This piece of text was written in the city of Auckland.`	`e:5 i:5 a:3 o:2 u:1`

Sample Input 3	Output for Sample Input
`ACM Programming Contest.`	`a:2 o:2 e:1 i:1 u:0`

I • Brocard Point of a Triangle

The *Brocard point* of a triangle ABC is a point P in the triangle chosen so that:
$\angle PAB = \angle PBC = \angle PCA$ (see figure below).



The common angle is called the *Brocard angle*. The largest *Brocard angle* is $\pi/6$ which is the *Brocard angle* for an equilateral triangle (the *Brocard point* is the centroid of the triangle).

Write a program to compute the coordinates of the *Brocard point* of a triangle given the coordinates of the vertices.

Input

The first line of input contains a single integer P, $(1 \le P \le 10000)$, which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of a single line of input. It contains the data set number, ***K,*** followed by the six space separated coordinate values `Ax`, `Ay`, `Bx`, `By`, `Cx`, `Cy` of the vertices of the triangle. The vertices will always be specified so going from `A` to `B` to `C` and back to `A` circles the triangle counter-clockwise. Input coordinates are floating point values.

Output

For each data set there is a single line of output. The single output line consists of the data set number, ***K***, followed by a single space followed by the ***x*** coordinate of the *Brocard point*, followed by a single space followed by the ***y*** coordinate of the *Brocard point*. Coordinates should be rounded to five decimal places.

Sample Input	Sample Output
3 1 0 -1.3 3.4 0.5 1.1 2.3 2 0 0 3 0 0 4 3 3.1 0.2 4.3 0.4 0 0.8	1 1.40456 0.82890 2 1.56047 0.74902 3 3.87699 0.40167

Problem J

Jumping Yoshi

Yoshi is a frog. He lives in the ZOO under a log which was specially transported there from a distant equatorial rainforest. The log is big and wet and attracts the flies and that is what makes Yoshi satisfied.

There is also a line of pebbles which runs through the wetland in front of the log. There are various dark spots on the pebbles and sometimes Yoshi likes to look at them and dream that they are not just spots but some really big flies instead.

Yesterday, Yoshi's friend camel Addawser came to see him and suggested to play a game.

"Do you see those spots on the pebbles?" asked Addawser. "I challenge you to start on the leftmost pebble and then do some jumps from a pebble to another pebble but with some restrictions. You can jump from a pebble to another one only if the sum of numbers of spots on both pebbles is equal to the distance between the pebbles. And you have to get to as distant pebble as possible."

"All right, but you know that I can count at most to twenty three and no more," hesitated Yoshi.

"No problem, I will help you with the bigger numbers," said Addawser.

"Then, it's easy," said Yoshi, positioning himself on the first pebble and looking inquisitively over the rest of the line. "Although, one might not be quite so sure, after all," he added after a while.

You are given the sequence of numbers of dark spots on the pebbles in the line. You are asked to find the most distant pebble which can be reached by a sequence of jumps. The first jump starts at the first pebble in the line and a jump between two pebbles is possible if and only if the sum of numbers of spots on both pebbles is equal to the distance between them. You may suppose that the line of pebbles is straight and that the distance between each two neighboring pebbles is exactly one frog distance unit.

Input Specification

There are more test cases. Each case starts with a line containing one integer N ($1 \leq N \leq 1\,000\,000$) representing the number of pebbles. The second line contains a list of N integers. The order of the integers in the list is the same as the order of the pebbles in the wetland, each integer represents the number of spots on the corresponding pebble. No pebble contains more than 10^9 spots. Suppose that Addawser knows all different pairs of pebbles where Yoshi can perform a jump from one pebble to another one during his sequence of jumps. You are guaranteed that the number of those pairs of pebbles never exceeds $1\,000\,000$.

The input is terminated by a line with one zero.

Output Specification

For each test case, print a single line with one integer denoting the distance of the pebble which can be reached by successive jumps according to the given rules and which is the most distant from the first pebble.

Sample Input

```
7
2 1 0 1 2 3 3
11
7 6 1 4 1 2 1 4 1 4 5
0
```

Output for Sample Input

```
5
10
```

K • Farey Sums

Given a positive integer, ***N***, the sequence of all fractions ***a*** / ***b*** with ($\boldsymbol{0} < \boldsymbol{a} \le \boldsymbol{b}$), ($\boldsymbol{1} < \boldsymbol{b} \le \boldsymbol{N}$) and ***a*** and ***b*** relatively prime, listed in increasing order, is called the *Farey Sequence of order **N***.

For example, the *Farey Sequence of order **6*** is:

0/1, **1/6**, **1/5**, **1/4**, **1/3**, **2/5**, **1/2**, **3/5**, **2/3**, **3/4**, **4/5**, **5/6**, **1/1**

If the denominators of the *Farey Sequence of order **N*** are:

b[***1***], ***b***[***2***], …, ***b***[***K***]

then the *Farey Sum* of order ***N*** is the sum of ***b***[***i***] / ***b***[***i***+**1**] from ***i*** = **1** to ***K*** –**1.**

For example, the *Farey Sum of order **6*** is:

1/6 + **6/5** + **5/4** + **4/3** + **3/5** + **5/2** + **2/5** + **5/3** + **3/4** + **4/5** + **5/6** + **6/1** = **35/2**

Write a program to compute the *Farey Sum of order **N*** (input).

Input

The first line of input contains a single integer P, ($1 \le P \le 10000$), which is the number of data sets that follow. Each data set should be processed identically and independently.

Each data set consists of a single line of input. It contains the data set number, K**,** followed by the order ***N*****,** N ($2 <= N <= 10000$), of the *Farey Sum* that is to be computed.

Output

For each data set there is a single line of output. The single output line consists of the data set number, K, followed by a single space followed by the *Farey Sum* as a decimal fraction in lowest terms. If the denominator is **1**, print only the numerator.

Sample Input	Sample Output
4 1 6 2 15 3 57 4 9999	1 35/2 2 215/2 3 2999/2 4 91180457/2

Problem L

Lunch Menu

Willy is the youngest zookeper employed in the ZOO. His income is not exactly a billionaire's one and he obviously has to plan his regular expenses carefully. Take for example his daily visit to the ZOO cantine. From the very beginning of his service in the ZOO, Willy decided that his daily lunch expense will not exceed a fixed limit L. And while his budget is limited he still wants to have a complete lunch: A soup, a main dish, a dessert, and a beverage. Moreover, to keep himself amused, he wants to enjoy each day a different menu, different from all other menus he had eaten in the previous days. Now, he wonders how many days will it take until he is forced to eat a lunch which is an exact copy of another lunch he had already eaten before.

You are given Willy's lunch price limit L and the prices of all soups, main dishes, desserts, and beverages in the cantine. Determine how many different lunches can be assembled provided that two lunches are different if they differ in at least one of the four given parts.

Input Specification

There are more test cases. Each case starts with a line containing five integers L, S, M, D, B N ($1 \leq L \leq 10^8$, $1 \leq S, M, D, B \leq 5\,000$) representing (in this order) the lunch price upper limit, the number of soups, the number of main dishes, the number of desserts and the number of beverages in the cantine. Each of the next four lines contains a list of prices. The first line contains the soups price list, the second line contains the main dishes price list, the third line contains the desserts price list, and the fourth line contains the beverages price list. All items in all lists are positive integers not exceeding 10^8. There is one empty line after each test case. The input is terminated by a line with five zeros.

Output Specification

For each test case print on a separate line the number of different lunches which can be assembled from the cantine offer and have price not exceeding L. Please note that the value of the solution might not fit into 32-bit integer.

Sample Input

```
11 3 1 1 1
4 5 6
3
2
1

10 4 5 4 2
3 2 5 7
1 1 8 4 2
3 5 2 1
2 3

0 0 0 0 0
```

Output for Sample Input

```
2
48
```