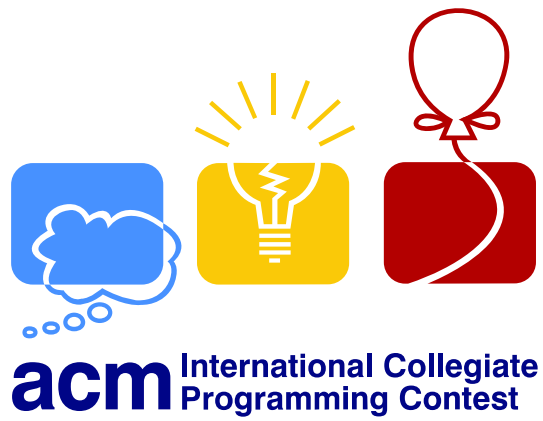


FAU Local Summer Contest

2004



The Problem Set

No	Title	Page
A	Aragorn	3
B	Mate in ??? moves	5
C	The Good Old Times	7
D	The Vase	9
E	Google	11
F	Number Theory	13
G	Continued Fractions	15
H	Cancelled Flights	17

Good luck and have fun!

This page is intentionally left (almost) blank.

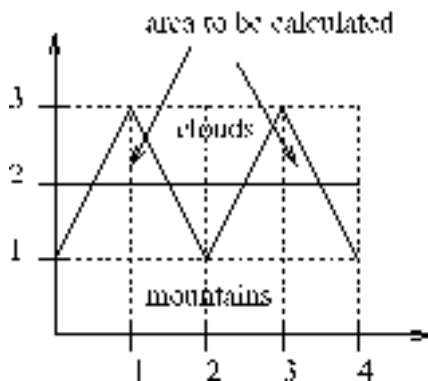
Problem A

Aragorn

This problem takes you back in time to Middle Earth, shortly before Sauron gets overturned. Here your task is to help Aragorn, also known as Strider, to successfully lead an army of elves over the Misty Mountains. Unfortunately the weather, which has been fair for the last few days, has turned nasty. To cross the Misty Mountains the army must reach the pass of doom. This wouldn't be a problem if the skies were clear, but today it is very cloudy. Therefore an army of orcs may be hiding in the clouds. As the chief counselor of Aragorn your task is to help decide, if it is safe to lead the army to the pass. This depends on how much of the mountains is covered with clouds.

Input

The input consists of several testcases. In each testcase you will be given a description of the clouds and a description of the mountains. Both will be given as a series of connected line segments, the first series of line segments describing the clouds, the second the mountains. Everything "above" the first series of line segments is covered with clouds. Everything "below" the second series of line segments belongs to the silhouette of the mountains, as shown in the following picture.



The series of line segments describing the clouds and the mountains will be given as series of points. Two neighbouring points form a line segment. Each point consists of a x -coordinate and y -coordinate. The first number in each test case is an integer c , $c \leq 1000$, denoting the number of points describing the clouds, followed by c pairs of floating point numbers, each describing a point. The next number is an integer m , $m \leq 1000$, denoting the number of points in the silhouette of the mountains, followed by m pairs of floating point numbers, each describing a point. Each sequence of line segments will begin at $(0, y)$ and end at the same x -coordinate as the other sequence. The x -coordinates of the points will grow steadily. Each y -coordinate will be greater than or equal to zero. You should parse until $c = 0$.

Output

You are to calculate how big an area of the mountains is covered by the clouds. Output the area covered by the clouds. The number should contain two digits after the decimal point.

Sample Input

```
2
0.0 3.0
4.0 3.0
5
0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0

2
0.0 2.0
4.0 2.0
5
```

0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0

2
0.0 0.0
4.0 0.0
5
0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0

2
0.0 0.0
4.0 0.0
5
0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0

5
0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0
3
0.0 2.0
2.0 1.0
4.0 2.0

3
0.0 2.0
2.0 1.0
4.0 2.0
5
0.0 1.0
1.0 3.0
2.0 1.0
3.0 3.0
4.0 1.0

0

Sample Output

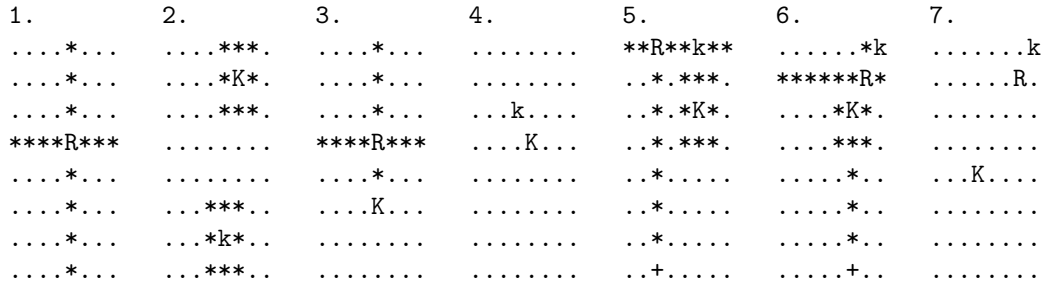
0.00
1.00
8.00
8.00
0.40
2.40

Problem B

Mate in ??? moves

Some people know chess. Chess is a computationally very difficult problem. The problem here is a little bit easier. Given a chess position consisting of a white rook R, a white king K and a black king k, you must compute the minimum number of moves needed to checkmate black, assuming white and black play optimally. White always begins.

The way the rook moves and the king moves is shown in the first two diagrams. The squares reachable by the rook and the king are marked with stars. The rook cannot jump over his own king as shown in the third diagram. Two kings may never stand next to each other as shown in the fourth diagram. If the black king is attacked by the white rook and cannot escape to a square which isn't attacked by the white rook and the white king, as shown in diagram 5, the black king is mated and black loses. If it is black's move in diagram 6, the game ends in a draw because black cannot move, even though it is his turn. Every square black could move to is attacked by the rook. He cannot capture the rook because it is guarded by the white king. In diagram 7 black can capture the white rook. If only two kings remain on the board the game ends in a draw.



Input

The first number in the input will be an integer t on a single line, giving the number of testcases. A testcase consists of a number n on a single line, $4 \leq n \leq 8$, which is the size of the chessboard. The next n lines contain a description of the board. Each line contains n characters. A . indicates an empty field, a R the white rook, a K the white king and a k the black king. All initial positions will be valid, so initially the two kings aren't attacking each other and the white rook isn't attacking the black king. There will be no trailing spaces or blank lines in the input.

Output

The output consists of one number. The number should be the minimum number of moves needed to checkmate black, assuming that both black and white play optimally. Here the movement of a single piece counts as a move. You may assume black can be mated in less than 10 moves. There should be no blank lines between the numbers.

Sample Input

```

2
8
.....k.
..R.....
.....K.
.....
.....
.....
.....
.....
.....
8
.....k.
.....
.....K.
.....R..
.....
.....

```

.....
.....

Sample Output

1
3

Problem C

The Good Old Times

In the stoneage things weren't as easy for programmers as they are today. For example programmers had only very slow computers with a very limited main memory and very small hard disks at their disposal. Furthermore a lot of standard applications hadn't been developed. Now one of your customers, Fred Flintstone, wants a command line calculator for his computer, the new Granite500 with 1000 hertz. Your task is to write a command line calculator for him.

Input

Each line of input will consist of a string of length $l, l < 255$, containing a valid arithmetic expression. Because main memory was very small in those days, the string will contain no blanks or tabs, nor parathensis. It will contain the four standard arithmetic operators $+, -, *, /$ as well as a unary $-$ or $+$ and floating point numbers. The input will be terminated by EOF.

Output

For each line of input, output the value of the arithmetic expression on a single line. The result should contain three digits after the decimal point.

Sample Input

```
1/2/2
-3.0
3
4.0+3.0/5.0
1*2*3+1+1*2+1*2*3*4
```

Sample Output

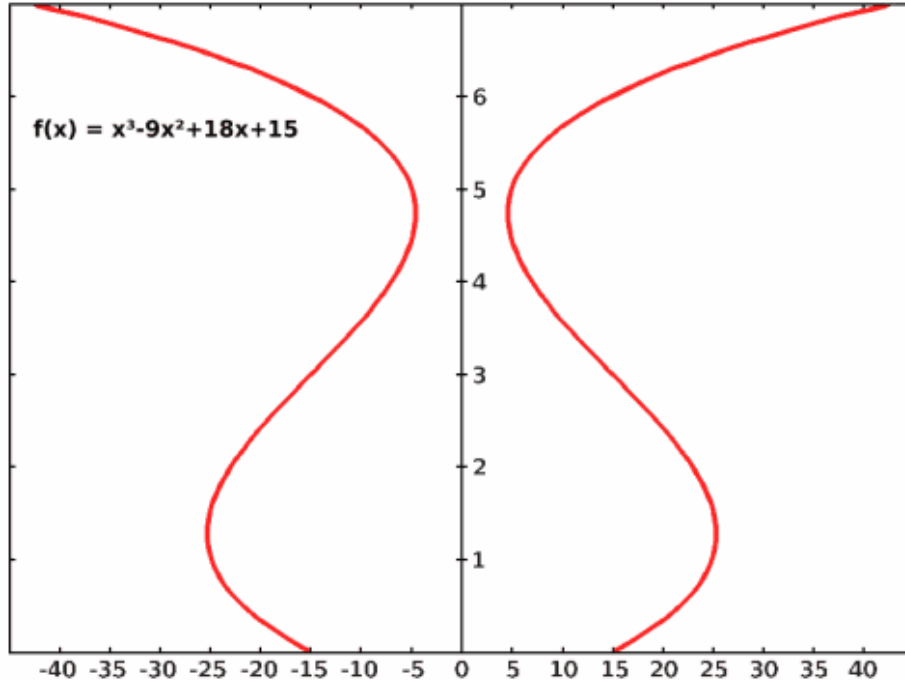
```
0.250
-3.000
3.000
4.600
33.000
```

This page is intentionally left (almost) blank.

Problem D

The Vase

Gordon Oliver Mattrey is a hobby mathematician. His wife has recently bought some flowers (as he does not do such things) and put them into a vase. Gordon is interested in the amount of water this vase can hold. Unfortunately, this calculation does not seem to be easy. He has, however, the assumption that the shape of the vase corresponds to a polygon of degree 3 (i.e. if x is the height of a certain point, we can express the radius of the vase at that point by $f(x)$, where f is a polynomial of degree at most 3). Given this information, the height of the vase and its diameter at 4 distinct points, can you help G. O. Mattrey and calculate the volume of the vase?



The picture above shows the vase that corresponds to the polynomial $f(x) = x^3 - 9 \cdot x^2 + 18 \cdot x + 15$. This vase is described in the first test case of the Sample Input.

Input

The first line of the input contains n , the number of test cases that follow. Each of the next n lines contains 9 numbers: h (the total height of the vase in cm) and $(h_1, d_1) \dots (h_4, d_4)$ (the height and diameter of the vase at 4 distinct measuring points, each in cm). None of these values will be bigger than 100.

Output

For every test case, print one line containing the volume of the vase in cm^3 , rounded to the nearest integer.

Sample Input

```
3
7.0 0.0 30 3.0 30 6.0 30 7.0 86
4.0 0.5 4 1.5 4 2.5 4 3.5 4
10 1 20 7 20 3 20 5 20
```

Sample Output

```
7827
50
3142
```

This page is intentionally left (almost) blank.

Problem E

Google

Efficiently searching the web is getting an increasingly important topic. Today, you have to write a simple Google. Given a set of documents (simple strings), your program has to answer queries of the form `string1 string2 string3` by returning a list of documents that match the query along with the number of occurrences of query strings in these documents.

The documents have to be sorted in descending order of query string occurrences.

An Example

We have three documents:

1. `"this_is_a_document_that_contains_many_underscores"`
2. `"_this_is_another_document_which_contains_even_more_underscores_"`
3. `"thethirddocumentdoesnotcontainanyunderscoresatall"`

Working on these documents, your program should return

Document 1: 2 matches

Document 2: 2 matches

Document 3: 2 matches

for the query `underscore document`, since all of the above documents contain one occurrence of "underscore" and one occurrence of "document".

For the query `is_ even_more`, your program should return

Document 2: 3 matches

Document 1: 2 matches

since the third document does not contain any of the query strings, the first document contains two occurrences of the string "is_", and only the second document contains both "is_" (2x) and "even_more" (1x).

Input

The first line of the input contains t , the number of test cases that follow. The first line of each test case consists of two integer numbers, $d < 20$ (the number of documents in this test case) and $q < 200$ (the number of queries to be processed). Each of the next $\lfloor d/i \rfloor$ lines contains a string, the j 'th document of this test case. Each of the remaining q lines consists of one integer $0 < n_i < 20$, the number of strings in the query, followed by n_i strings.

No string in the input will contain a whitespace character. No string will be longer than 10000 characters.

Output

For every query in every test case, print one line for each document that contains at least one occurrence of any string found in the query (so, at most d lines, where d is the number of documents in that test case). Use the format shown in the example above and in the Sample Output. Sort the documents that match the query in descending order of the number of occurrences. If there are several documents with the same number of occurrences, print the one first that appeared earlier in the input (like in the example above). If no document matches the query, print one line: "No document found."

Print a blank line after each query. Please take care of "1 match" and "2 matches".

Sample Input

```
2
3 4
this_is_a_document_that_contains_many_underscores
_this_is_another_document_which_contains_even_more_underscores_
thethirddocumentdoesnotcontainanyunderscoresatall
2 underscore document
2 is_ even_more
3 does not contain
2 contain any
```

```
1 2
blablabla
3 bla bla blabla
1 blahblah
```

Sample Output

```
Document 1: 2 matches
Document 2: 2 matches
Document 3: 2 matches
```

```
Document 2: 3 matches
Document 1: 2 matches
```

```
Document 3: 3 matches
Document 2: 2 matches
Document 1: 1 match
```

```
Document 1: 2 matches
Document 3: 2 matches
Document 2: 1 match
```

```
Document 1: 8 matches
```

```
No document found.
```

Problem F

Number Theory

Mathematicians are a curious breed of people. Especially number theorists. They spend most of their time thinking about different properties of numbers. Albert Meyer, a number theorist, is trying to discover an interesting sequence of positive integers. He suspects the sequence i_1, i_2, i_3, \dots in which the value of i_n is the number of numbers $m, 1 \leq m \leq n$, where $\gcd(m, n) = 1$ and $\gcd(m, n) = m$, is very interesting. gcd is short for "greatest common divisor". He has turned to you, as you are an expert programmer and the calculations by hand are very tedious, to calculate a few numbers in this sequence.

Input

The input will consist of several positive integers $0 < n < 2^{31}$. The input will be terminated by EOF.

Output

For each number output the number of numbers $m, 1 \leq m \leq n$, where $\gcd(m, n) = 1$ and $\gcd(m, n) = m$.

Sample Input

```
1
2
6
2147000000
```

Sample Output

```
0
0
1
1340599805
```

This page is intentionally left (almost) blank.

Problem G

Continued Fractions

Every rational number can be expressed as a so-called finite "continued fraction" (german: "Kettenbruch"), an expression of the following form, where all $q_i (0 \leq i \leq n)$ are integers:

$$x = q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{\dots + \frac{1}{q_n}}}}$$

This representation is unique, if we require q_1, \dots, q_n to be non-negative and if $n > 0$ then $q_n > 1$. Your task will be to convert rational numbers given as quotients a/b (where $a \geq 0, b > 0$ are integers) to its continued fraction expansion, written as $[q_0, q_1, q_2, \dots, q_n]$. Here q_0 is an integer and for $n > 0, q_n$ is greater than 1 and all q_i are non-negative integers ($1 \leq i \leq n$). If $q_n = 1$ then with $Q_{n-1} := q_{n-1} + 1/q_n = q_{n-1} + 1$ by $[q_0, \dots, Q_{n-1}]$, we obtain the same number as by $[q_0, \dots, q_{n-1}, q_n]$, because $q_{n-1} + 1 = q_{n-1} + 1/1 = q_{n-1} + 1/q_n$.

Input

The input consists of several lines. Each of these lines will contain one quotient a/b and no whitespaces. Both numbers a and b will integers, $b > 0$. The last line of input will contain the fraction $0/0$, which should not be processed.

Output

For each quotient (except the final "0/0") print its representation as continued fraction $[q_0, q_1, \dots, q_n]$ on a separate line. q_0, \dots, q_n must be integers. Furthermore q_1, \dots, q_n should be non-negative and if $n > 0$, then q_n must be greater than 1.

Sample Input

```
355/113
-3/2
21/532
3/1
0/0
```

Sample Output

```
[3, 7, 16]
[-2, 2]
[0, 25, 3]
[3]
```

This page is intentionally left (almost) blank.

Problem H

Cancelled Flights

Saving money is nowadays an important issue for all firms. This is especially true for airlines as the passenger number have dropped in the last few years. So FAU airways is considering to cancel flights on some of their routes in order to save money. But they want to make sure that it is still possible to reach every airport from any other airport by flying with one of their airplanes. Your task is to help FAU airways to determine which routes can be cancelled.

Input

The first line of the input gives you the number of airports FAU airways is flying to, the second line gives the number of routes they are currently servicing. The following lines specify all the routes. A route consists of the three letter code of the first airport followed by space a dash another space and the three letter code of the second airport. The letters are in the range from A-Z and all routes are serviced in both directions.

Output

Next the number of routes that are considered for cancellation is given in a new line, followed by the actual routes in the same format as above. Your program should output **no** if the route can not be cancelled and **yes** if after cancellation it is still possible to reach every airport from any other airport by flying FAU airways. Every route is considered for itself, so no route is actually removed if it could be.

Sample Input

```
6
7
AAA - BBB
AAA - CCC
CCC - DDD
DDD - EEE
DDD - FFF
EEE - FFF
BBB - CCC
3
AAA - BBB
CCC - DDD
EEE - FFF
```

Sample Output

```
yes
no
yes
```

This page is intentionally left (almost) blank.