

Al-Khawarizmi 09 Programming Competition

Hosted by International Islamic University
Malaysia



22nd and 23rd November, 2009

**You get 16 Pages
10 Problems
&
300 Minutes**



A

Numbering Roads!

Input: Standard Input
Output: Standard Output



In my country, streets don't have names, each of them are just given a number as name. These numbers are supposed to be unique but that is not always the case. The local government allocates some integers to name the roads and in many case the number of integers allocated is less that the total number of roads. In that case to make road names unique some single character suffixes are used. So roads are named as 1, 2, 3, 1A, 2B, 3C etc. Of course the number of suffixes is also always limited to 26 (A, B, ..., Z). For example if there are 4 roads and 2 different integers are allocated for naming then some possible assignments of names can be:

1, 2, 1A, 2B
1, 2, 1A, 2C
3, 4, 3A, 4A
1, 2, 1B, 1C

Given the number of roads (**R**) and the numbers of integers allocated for naming (**N**), your job is to determine minimum how many different suffixes will be required (of all possible namings) to name the streets assuming that no two streets can have same names.

Input

The input file can contain up to **10002** lines of inputs. Each line contains two integers **R** and **N** ($0 < N, R < 10001$). Here **R** is the total number of streets to be named and **N** denotes number integers allocated for naming.

Output

For each line of input produce one line of output. This line contains the serial of output followed by an integer **D** which denotes the minimum number of suffixes required to name the streets. If it is not possible to name all the streets print "impossible" instead (without the quotes).

Sample Input

```
8 5
100 2
0 0
```

Output for Sample Input

```
Case 1: 1
Case 2: impossible
```

Problem Setter: Shahriar Manzoor, Special Thanks: Sohel Hafiz



B

Evaluate the Expression

Input: Standard Input
Output: Standard Output

In this problem, we will consider a mathematical expression according to the following BNF grammar:

```
<expression> = <variable> | <expression><operator><expression> |
"("<expression>")"
<operator> = "+" | "*"
<variable> = "a" | "b" | "c" | ... | "y" | "z"
```

When evaluating such expressions, you have to follow the conventional rules. That means you have to do things in the brackets first and multiplications have to be done before addition.

Example: $2 * (3 + 4 * 2) = 22$

Given an expression and some inequalities, you have to assign each variable with a positive integer so that the value of the expression is minimized.

Consider an example:

Expression = $a + b * c$ and Inequalities = $a > b, c > b$

Assignment of: $a=2, b=1$ and $c=2$ will give us the minimum value. $\Rightarrow 2 + 1 * 2 = 4$

Input

The first line of input is an integer $T(T < 100)$ that gives us the number of test cases. Each case starts with a line that gives you the expression. The length of the expression is at most 300. The next line contains an integer $I(I < 400)$ that gives you the number of inequalities. Each of the next I lines will give you an inequality of the format $x > y$ where x and y are lowercase alphabets that are present in the given expression and x is not equal to y . All the inequalities will be distinct.

Output

For each case, output the case number first. Then output the minimum value of the expression that can be obtained by assigning positive integers to each variable that abides by the given inequalities. You can assume the output will fit in 32 bit signed integer. If the inequalities are inconsistent, then output -1 instead.

$$a = x$$

$$a + a = a + x$$

$$2a = a + x$$

$$2a - 2x = a + x - 2x$$

$$2(a-x) = a + x - 2x$$

$$2(a-x) = a - x$$

$$2 = 1$$





Sample Input

```
3
a+b*c
2
a>b
c>b
z*(x+y)
3
z>x
x>y
z>y
a+b+c+a
0
```

Output for Sample Input

```
Case 1: 4
Case 2: 9
Case 3: 4
```

Problem Setter: Sohel Hafiz, Special Thanks: Renat Mullakhanov



C

Colorful Board

Input: Standard Input
Output: Standard Output

You are given a board. You are asked to draw M horizontal lines and N vertical lines in that board, so that the whole board is divided into $(M+1) \times (N+1)$ cells. So there will be $M + 1$ rows each of which will exactly contain $N + 1$ cells.

Now you are asked to color every cell. For these purpose, you are given four colors.

Red
Green
Blue
Orange

To make the board more beautiful and colorful, you have to be careful, so that no two adjacent cells contain the same color. Two cells are considered adjacent if they share a common side.

You are also given some forbidden cells. You must not draw anything in those cells. But you must color every other cell which are not forbidden using a single color.

You have to answer in how many ways you can color this board.



Figure: Some examples of valid and invalid coloring, where $M = 1$, $N = 1$ and no forbidden cell.

Input

Input will start with an integer T ($T \leq 50$) which indicates the number of test case. Each case will start with two integers M and N ($0 \leq M, N \leq 6$), number of horizontal lines and number of vertical lines. Next line will contain an integer K ($0 \leq K \leq (M+1) \times (N+1)$), which indicates the number of forbidden cells. Each of the next K lines will contain two integers x and y ($1 \leq x \leq M+1, 1 \leq y \leq N+1$), representing one forbidden cell, which is y th cell of x th row. Each given forbidden cells are distinct.

Output

For each test case, output a single line in the form "Case #: P", where # will be replaced by the case number and P will be replaced by the number of valid ways you can draw the given board. The result can be very large you should output the result modulo 1000000007.



Sample Input

```
2
1 1
1
2 1
0 0
0
```

Output for Sample Input

```
Case 1: 36
Case 2: 4
```

Problem Setter: Md. Arifuzzaman Arif, Special Thanks: Mohammad Mahmudur Rahman



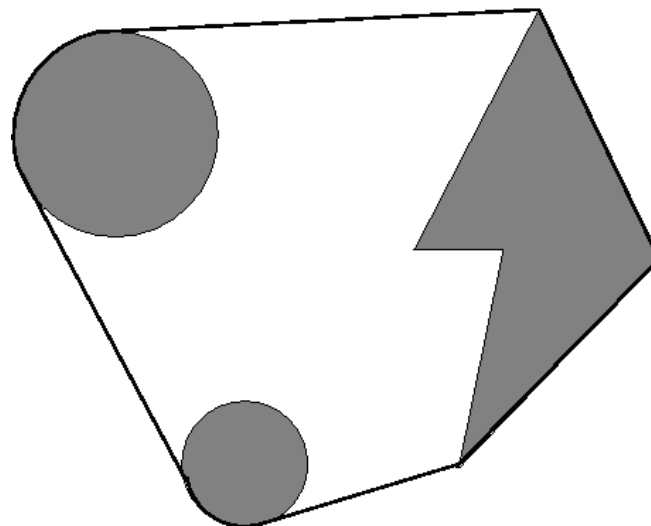
D

Crime Scene

Input: Standard Input
Output: Standard Output

You have been appointed as a programmer in your local police station (they watch a lot of TV shows like CSI, Numb3rs etc., and though having a computer genius among them would be pretty cool). They have given you a desktop, Now, you can do all the “impossible” tasks, that are being done in the shows frequently (like magnifying an image infinite times with no pixelation, or building systems, where the only way to reboot is to shut down the main power of the building).

But due to the recent increase in crime and also the economic problems, the station is planning to make some budget cuts. One of the cut is in the use of crime scene tapes. So, at each crime scene, they want to use the minimal possible amount of tape.



Two circles and a polygon in the crime scene. The optimal taping is shown in dark.

You are given all the objects in the crime scene. The objects are all circular or in the shape of a polygon. Return the minimum amount of crime scene tapes to surround them.

Input

First line contains T , the number of test cases. Each test case starts with N , the number of objects in the crime scene. Each of the next N lines describe an object in the crime scene. The first character in the line will be either 'c' or 'p', denoting whether the object is a circle, or a polygon. If it is a circle, then, it will be followed by 3 real numbers, (cx, cy) the center of the circle and radius of the circle.

In case of polygon, it will be followed by an integer K , the number of sides the polygon will have, followed by K pairs of real numbers, (x_i, y_i) the location of the i -th vertex of the polygon.

All polygons in this problem will be simple.



Output

For each test case, output the case number and the optimal length of the tape. Output 6 digits after the decimal point.

Constraints

- $N \leq 100$
- $K \leq 10$
- All radii will be greater than 0.
- All input values will be between -1000 and +1000.

Sample Input

Output for Sample Input

1 2 c 0 0 2 p 4 -1 2 1 2 1 -2 -1 -2	Case #1: 13.148009
--	--------------------

Problem Setter: Manzurur Rahman Khan, Special Thanks: Samee Zahur



E

Cost Cutting

Input: Standard Input
Output: Standard Output

Company XYZ have been badly hit by recession and is taking a lot of cost cutting measures. Some of these measures include giving up office space, going open source, reducing incentives, cutting on luxuries and issuing pink slips.

They have got three (3) employees working in the accounts department and are going to lay-off two (2) of them. After a series of meetings, they have decided to dislodge the person who gets the most salary and the one who gets the least. This is usually the general trend during crisis like this.

You will be given the salaries of these 3 employees working in the accounts department. You have to find out the salary of the person who survives.



Input

The first line of input is an integer **T** ($T < 20$) that indicates the number of test cases. Each case consists of a line with 3 distinct positive integers. These 3 integers represent the salaries of the three employees. All these integers will be in the range [1000, 10000].

Output

For each case, output the case number followed by the salary of the person who survives.

Sample Input

```
3
1000 2000 3000
3000 2500 1500
1500 1200 1800
```

Output for Sample Input

```
Case 1: 2000
Case 2: 2500
Case 3: 1500
```

Problem Setter: Sohel Hafiz, Special Thanks: Md. Arifuzzaman Arif



F

Alternate Task

Input: Standard Input
Output: Standard Output

Little Hasan loves to play number games with his friends. One day they were playing a game where one of them will speak out a positive number and the others have to tell the sum of its factors. The first one to say it correctly wins. After a while they got bored and wanted to try out a different game. Hassan then suggested about trying the reverse. That is, given a positive number S , they have to find a number whose factors add up to S . Realizing that this task is tougher than the original task, Hasan came to you for help. Luckily Hasan owns a portable programmable device and you have decided to burn a program to this device. Given the value of S as input to the program, it will output a number whose sum of factors equal to S .

Input

Each case of input will consist of a positive integer $S \leq 1000$. The last case is followed by a value of 0 .

Output

For each case of input, there will be one line of output. It will be a positive integer whose sum of factors is equal to S . If there is more than one such integer, output the largest. If no such number exists, output -1 . Adhere to the format shown in sample output.

Sample Input

```
1
102
1000
0
```

Output for Sample Input

```
Case 1: 1
Case 2: 101
Case 3: -1
```

Problem Setter: Shamim Hafiz, Special Thanks: Sohel Hafiz



G

Commando War

Input: Standard Input
Output: Standard Output

*“Waiting for orders we held in the wood, word from the front never came
By evening the sound of the gunfire was miles away
Ah softly we moved through the shadows, slip away through the trees
Crossing their lines in the mists in the fields on our hands and our knees
And all that I ever, was able to see
The fire in the air, glowing red, silhouetting the smoke on the breeze”*

There is a war and it doesn't look very promising for your country. Now it's time to act. You have a commando squad at your disposal and planning an ambush on an important enemy camp located nearby. You have N soldiers in your squad. In your master-plan, every single soldier has a unique responsibility and you don't want any of your soldier to know the plan for other soldiers so that everyone can focus on his task only. In order to enforce this, you brief every individual soldier about his tasks separately and just before sending him to the battlefield. You know that every single soldier needs a certain amount of time to execute his job. You also know very clearly how much time you need to brief every single soldier. Being anxious to finish the total operation as soon as possible, you need to find an order of briefing your soldiers that will minimize the time necessary for all the soldiers to complete their tasks. You may assume that, no soldier has a plan that depends on the tasks of his fellows. In other words, once a soldier begins a task, he can finish it without the necessity of pausing in between.

Input

There will be multiple test cases in the input file. Every test case starts with an integer N ($1 \leq N \leq 1000$), denoting the number of soldiers. Each of the following N lines describe a soldier with two integers B ($1 \leq B \leq 10000$) & J ($1 \leq J \leq 10000$). B seconds are needed to brief the soldier while completing his job needs J seconds. The end of input will be denoted by a case with $N = 0$. This case should not be processed.

Output

For each test case, print a line in the format, “Case X: Y”, where X is the case number & Y is the total number of seconds counted from the start of your first briefing till the completion of all jobs.



Sample Input

```
3
2 5
3 2
2 1
3
3 3
4 4
5 5
0
```

Output for Sample Input

```
Case 1: 8
Case 2: 15
```

Problem Setter: Mohammad Mahmudur Rahman, Special Thanks: Manzurur Rahman Khan



H

Number Transformation

Input: Standard Input
Output: Standard Output

You are given an integer number **S**. You can transform any integer number **A** to another integer number **B** by adding **x** to **A**. This **x** is an integer number which is a prime factor of **A** (Please note that 1 and **A** are not being considered as a factor of **A**). Now, your task is to find the minimum number of transformations required to transform **S** to another integer number **T**.

Input

For each test case, there will be a line with two integers, **S** ($1 \leq S \leq 100$) & **T** ($1 \leq T \leq 1000$), as described above. The last test case will be followed by a line with two 0 s denoting end of output. This case should not be processed.

Output

For every test case except the last one, print a line of the form “Case X: Y” where **X** is the serial number of output (starting from 1). **Y** is the minimum number of transformations required to transform **S** to **T**. If it is not possible to make **T** from **S** with the given rules, **Y** shall be -1.

Sample Input

```
6 12
6 13
0 0
```

Output for Sample Input

```
Case 1: 2
Case 2: -1
```

Explanation of case 1: You can make 12 from 6 in 2 steps in this way: $6 \rightarrow 9 \rightarrow 12$.

Problem Setter: Mohammad Mahmudur Rahman, **Special Thanks:** Md. Arifuzzaman Arif



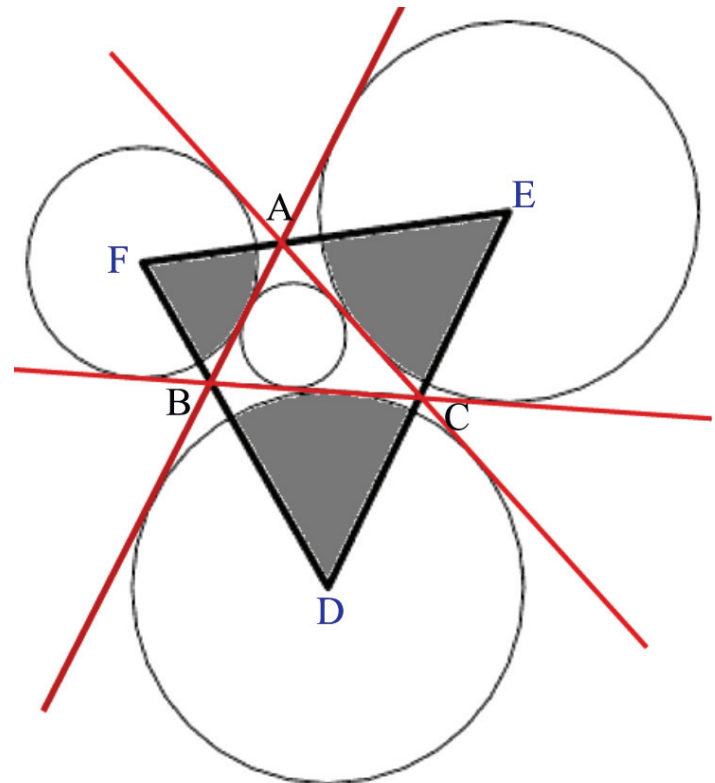
I

Ex-circles

Input: Standard Input
Output: Standard Output

In the figure on the right you can see triangle ABC and its in-circle (Circle that touches all the sides of a triangle internally) and three ex-circles (Circles that touch one side internally and other two sides externally). D, E and F are centers of the ex-circles.

Given the length of the sides of triangle ABC, you will have to find the area of triangle DEF and also the total area of the three grey shaded regions.



Input

The input file can contain up to 6000 lines of inputs. Each line contains three positive integer numbers a , b , c which denotes the length of the sides of the triangle ABC. You can assume that these three sides can form a valid triangle (positive area) and none of the side length is greater than 1000.

Input is terminated by a line containing three zeroes.

Output

For each line of input produce one line of output. This line contains the serial of output followed by two floating-point numbers. The first one denotes the area of triangle DEF and second one denotes the total area of the three grey shaded regions. This floating-point numbers should have two digits after the decimal point. You can assume that small precision errors will not cause difference in the printed output.

Sample Input

```
3 4 5
10 11 12
0 0 0
```

Output for Sample Input

```
Case 1: 30.00 21.62
Case 2: 211.37 144.73
```

Problem Setter: Shahriar Manzoor, **Special Thanks:** Sohel Hafiz



J

“strcmp()” Anyone?

Input: Standard Input
Output: Standard Output

strcmp() is a library function in C/C++ which compares two strings. It takes two strings as input parameter and decides which one is lexicographically larger or smaller: If the first string is greater then it returns a positive value, if the second string is greater it returns a negative value and if two strings are equal it returns a zero. The code that is used to compare two strings in C/C++ library is shown below:

```
int strcmp(char *s, char *t)
{
    int i;
    for (i=0; s[i]==t[i]; i++)
        if (s[i]=='\0')
            return 0;
    return s[i] - t[i];
}
```

Figure: The standard strcmp() code provided for this problem.

The number of comparisons required to compare two strings in strcmp() function is never returned by the function. But for this problem you will have to do just that at a larger scale. strcmp() function continues to compare characters in the same position of the two strings until two different characters are found or both strings come to an end. Of course it assumes that last character of a string is a null ('\0') character. For example the table below shows what happens when “than” and “that”; “therE” and “the” are compared using strcmp() function. To understand how 7 comparisons are needed in both cases please consult the code block given above.

t h a N \0	t h e r E \0
= = = ≠	= = = ≠
t h a T \0	t h e \0
Returns negative value	Returns positive value
7 Comparisons	7 Comparisons

Input

The input file contains maximum 10 sets of inputs. The description of each set is given below:

Each set starts with an integer N (0<N<4001) which denotes the total number of strings. Each of the next N lines contains one string. Strings contain only alphanumerals ('0'... '9', 'A'... 'Z', 'a'... 'z') have a maximum length of 1000, and a minimum length of 1.

Input is terminated by a line containing a single zero. Input file size is around 23 MB.



Output

For each set of input produce one line of output. This line contains the serial of output followed by an integer T. This T denotes the total number of comparisons that are required in the strcmp() function if all the strings are compared with one another exactly once. So for N strings the function strcmp() will be called exactly $\frac{N(N-1)}{2}$ times. You have to calculate total number of comparisons inside the strcmp() function in those $\frac{N(N-1)}{2}$ calls. You can assume that the value of T will fit safely in a 64-bit signed integer. Please note that the most straightforward solution (Worst Case Complexity $O(N^2 * 1000)$) will time out for this problem.

Sample Input

Output for Sample Input

2 a b 4 cat hat mat sir 0	Case 1: 1 Case 2: 6
---	------------------------

Problem Setter: Shahriar Manzoor, Special Thanks: Md. Arifuzzaman Arif, Sohel Hafiz, Manzurur Rahman Khan