Problem A : Pak Ga Nern

Input from file : `a.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

Pak and Ga are siblings from the Nern Family. They operate a small poultry business in a remote town.

One day, the siblings needed extra money to pay for their educational trip to the city. They went to their father and asked if they can have few baskets of eggs to sell. They were then given 2 baskets of eggs.

At the market, they set the baskets down. A few minutes later, a man pushing a full-load wheel-barrow accidentally stepped on their baskets, crushing some and cracking most of the eggs *but otherwise leaving them whole*. The man offered to pay for the damaged eggs and asked how many eggs they have. They could not tell. They have not counted the eggs in their baskets. So Ga took the crushed eggs out 3 at a time, there were 2 left in his basket. Pak also took the crushed eggs out from his basket 5 at a time, there were 3 left. They counted again 7 at a time. This time, both of them have 2 eggs left. So the man asked, "What could be the minimum number of eggs they could have?"

Your task is to help Pak and Ga Nern automate this task.

INPUT FORMAT

Input contains several lines. Each line contains three pairs of data separated by whitespaces. Each pair contains positive integers X and Y, where X is the number of eggs to be taken from the basket at a time and Y is the number of eggs left. Hence, the input takes the form:

$$X_1 Y_1 X_2 Y_2 X_3 Y_3$$

All X's are relatively prime and distinct across the data set.

OUTPUT FORMAT

For each line of input, display the smallest number the three pairs can produce. Otherwise, for cases not defined, display `malformed`.

CONSTRAINTS

$0 \leq Y < X < 1,000$

SAMPLE INPUT

```
3 2 5 3 7 2
10 4 11 1 15 2
```

SAMPLE OUTPUT

```
23
malformed
```

Problem B : Coefficients of a Polynomial

Input from file : `b.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 5 seconds

Let $p(x)$ be a polynomial of degree n, with unknown integer coefficients a_0, a_1, a_2, ... , a_n where

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n.$$

You are given the $n+1$ integer values $p(0)$, $p(1)$, $p(2)$, ... , $p(n)$ of the polynomial at the points $x = 0, 1, 2, \ldots, n$. Find the values of the coefficients a_0, a_1, a_2, ..., a_n.

INPUT FORMAT

The input consists of more than one data sets. The first line of each data set is an ID (at most six alphanumeric characters) followed by the value of n, the two separated by a space. This is followed by a line containing the $n + 1$ integer values $p(0)$, $p(1)$, $p(2)$, ... , $p(n)$ of the polynomial.

The next data set will immediately follow the previous data set.

A data set ID of "ZZ" will signify the end of input data.

OUTPUT FORMAT

For each data set, print the ID and the coefficients a_0 a_1 a_2 ... a_n of the polynomial.

CONSTRAINTS

$0 < n \leq 10$

$2,147,483,648 \leq p(i) \leq 2,147,483,647$ for $0 \leq i \leq 10$

The given values of $p(0)$, $p(1)$, ..., $p(n)$ will always define a $p(x)$ with integer coefficients.

SAMPLE INPUT

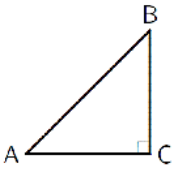
```
POLY01 2
5 10 19
POLY02 3
7 8 27 76
POLY03 4
5 3 9 47 165
ZZ
```

SAMPLE OUTPUT

```
POLY01 5 3 2
POLY02 7 -4 3 2
POLY03 5 -4 3 -2 1
```

Problem C : Billard Paths

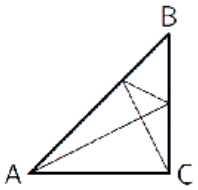
Input from file : `c.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

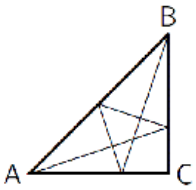


Let a point move on a frictionless plane bounded by a right isosceles triangle with vertices A, B, and C as shown. If the point hits a vertex, then it stops. If it hits a side, then it changes its direction of motion such that the angle of reflection is equal to the angle of incidence. The path that the point follows is called a billiard path.

Let a path start at vertex A with an initial slope $0 < m < 1$ chosen such that the path eventually ends at either vertex B or vertex C.

For example, for $m = 1/2$, the path ends at vertex C; for $m = 1/3$, the path ends at vertex B (as shown below).





INPUT FORMAT

The input is a list of integers separated by single spaces. The first integer is the number of paths p. The remaining integers are numerator-denominator pairs, n and d, that describe the initial slope of the paths. (For example, two paths with initial slopes $m_1 = 2/3$ and $m_2 = 1/2$ would be described by the input "`2 2 3 1 2`." The slopes are not necessarily given in lowest terms so the input "`2 4 6 3 6`" describes the same paths as the input "`2 2 3 1 2`.")

OUTPUT FORMAT

The output is a string composed of the characters B and/or C where each character represents the vertex where the corresponding path in the input ends (with the characters given in the same order as the corresponding paths).

CONSTRAINTS

$0 < p < 100$
$0 < n < 100$
$0 < d < 100$
$n < d$

SAMPLE INPUT

```
9 1 6 2 6 3 6 4 6 5 6 1 5 2 5 3 5 4 5
```

SAMPLE OUTPUT

```
CBCCCBCBC
```

Problem D : Counting Triangles

Input from file : `d.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

In a network of N nodes $\{n_0, n_1, \ldots, n_{N-1}\}$, a pair of nodes n_i and n_j are connected to each other if there does exist a link (i, j), for all possible pairs i and j.

A trio of nodes n_i, n_j, and n_k form a triangle if the links (i, j), (j, k), and (k, i) exist. This network is undirected so when a link (i, j) exists, so is the link (j, i). The problem is to count the number of such triangles given a network whose set of N nodes and set of M links are known. Note that only one triangle must be counted for any trio of nodes found. In other words, if the trio n_0, n_1, and n_3 form a triangle because $(0, 1)$, $(1, 3)$, and $(3, 0)$ exist, so is the trio n_0, n_3, and n_1 because $(0, 3)$, $(3, 1)$, and $(1, 0)$ also exist (see Figure 1 for an example). Both trios of nodes pertain to the same triangle and, thus, all such possible combinations must only be counted as one triangle.

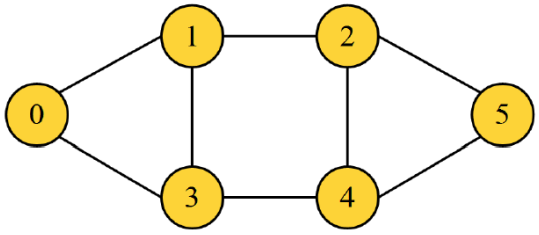


Figure 1: A network of $N = 6$ nodes (represented in this visualization as colored circles numbered from 0 to 5) and their links with each other (represented as lines connected to circle).

INPUT FORMAT

The input consists of several cases to be processed. The first line is a positive integer T, indicating the number of cases. The data for each of the T cases has the following specifications:

1. The first line of each case is a positive integer N which indicates the number of nodes in the network. The nodes are indexed as $\{0, 1, \ldots, N-1\}$.

2. The next line contains a positive integer M which means the number of links. Note that $M \leq N(N-1)$.

3. In the $M/2$ lines that follow, each contains a pair of non-negative integers i and j, where each pair is separated by at least one whitespace. Note that $i \neq j$, both must not be greater than $N-1$, and if (i, j) is explicitly defined in the file, then (j, i) is implicitly defined as well.

OUTPUT FORMAT

For each case in the input, print the corresponding number of triangles in one line.

CONSTRAINTS

$0 < T < 6$

SAMPLE INPUT

```
2
5
10
0 1
1 2
2 3
3 4
2 4
3
6
0 1
1 2
2 0
```

SAMPLE OUTPUT

```
1
1
```

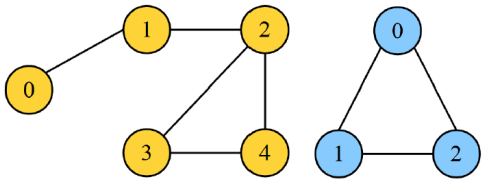


Figure 2: The visualization of the networks in the example input file. The first case has five nodes while the second has three. Both have a triangle count of one.

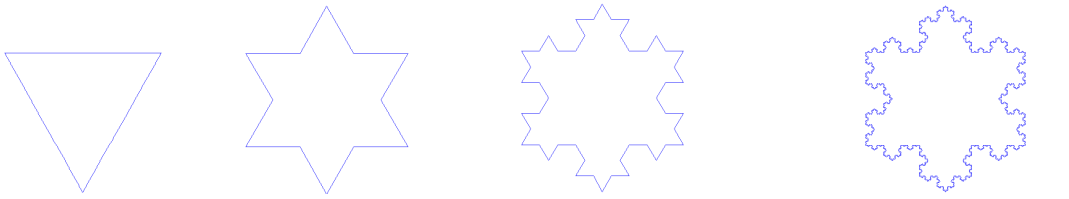
Problem E : Area of Koch Snowflake

Input from file : `e.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

The Koch snowflake is constructed starting from an equilateral triangle. In each iteration, each of the line segments of figure is altered as follows.

1. Divide the line segment into three segments of equal length.
2. Draw an equilateral triangle that has the middle segment from step 1 as its base and points outward.
3. Remove the line segment that is the base of the triangle from step 2.

After one iteration of this process, the resulting shape is the outline of a hexagram.



Starting figure After one iteration After two iterations ··· After six iterations

Your task is given the area A of the starting equilateral triangle, compute the area of the Koch snowflake after n iterations.

INPUT FORMAT

The input starts with a positive integer T, followed by T test cases. Each test case is a pair of positive integers A and n where A is the area of the starting equilateral triangle and n as the number of iterations.

OUTPUT FORMAT

For each test case, output the area of the nth iteration Koch snowflake given A, the area of the starting equilateral triangle. The area should be precise up to 10 decimal places after the decimal point.

CONSTRAINTS

$0 < A < 1,000,000$
$0 < n < 1,000$

SAMPLE INPUT

```
2
10 2
25 5
```

SAMPLE OUTPUT

```
14.8148148154
39.7398770515
```

Problem F : Pak Ga Nern, Level Up!

Input from file : `f.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

Alright! Pak and Ga congratulate you for solving the earlier problem. Curious as to how good your programming skill is, the Nern brothers have given you a challenge. They want you to solve the same problem given N number of data pairs.

INPUT FORMAT

Input contains several lines. Each line starts with a positive number N followed by N pairs of data separated by whitespaces. Each pair contains positive integers X and Y, where X is the number of eggs to be taken from the basket at a time and Y is the number of eggs left. Hence, the input takes the form

$$NX_1Y_1X_2Y_2 \cdots X_NY_N$$

All X's are relatively prime and distinct across the data set.

OUTPUT FORMAT

For each line of input, display the smallest number the N pairs can produce. Otherwise, for cases not defined by the problem, display `malformed`.

CONSTRAINTS

$0 \leq Y < X < 1,000$
$1 < N \leq 100$

SAMPLE INPUT

```
3 3 2 5 3 7 2
3 10 4 11 1 15 2
4 11 4 7 1 29 2 5 3
2 89 10 5 1
```

SAMPLE OUTPUT

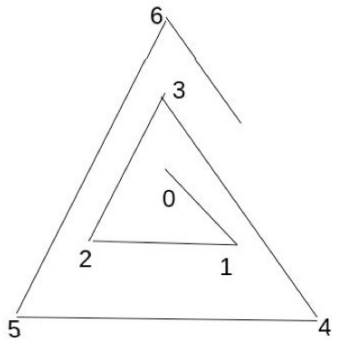
```
23
malformed
3018
366
```

Problem G : Triangular Spiral

Input from file : `g.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

A triangular spiral is formed by connecting the points p_0, p_1, p_2, ..., using straight line segments, as shown in diagram. Here the $(x, y)-$coordinates of the points are as follows:

$p_0 = (0, 0)$, $p_1 = (1, -1)$, $p_2 = (-1, -1)$, $p_3 = (0, 1)$, $p_4 = (2, -2)$, $p_5 = (-2, -2)$, $p_6 = (0, 3)$, $p_7 = (3, -3)$, $p_8 = (-3, -3)$, etc.



Given a nonnegative integer M, write a computer program that computes `seg`(M) and `len`(M). Here `seg`(M) is the length of the segment that connects $p_{(M-1)}$ to p_M. Also `len`(M) is the total length of the polyline that connects p_0 to p_1 to p_2 to ... to p_M. For example when $M = 3$, `seg`$(3) = 2.236$ approximately, and `len`$(3) = 1.414 + 2.000 + 2.236 = 5.650$ approximately.

INPUT FORMAT

The input consists of several data sets. Each data set consists of a single line of input that starts with a data set ID (at most six alphanumeric characters) followed by the value of M, the two separated by a space. The value of M will be between 0 and 10000. The next data set will immediately follow the previous data set. A data set `ID` of "`ZZ`" will signify the end of input data.

OUTPUT FORMAT

For each data set, print the data set `ID`, and the values of $A =$`seg`(M) and $B =$`len`(M) in this format:

`ID` A B

The values A and B must be correct to three decimal places.

CONSTRAINTS

$0 < M \leq 2,147,483,647$

SAMPLE INPUT

```
TS01 8
TS02 234
TS03 5432
TS04 10000
ZZ
```

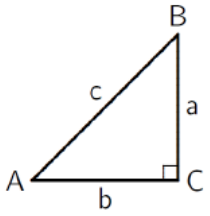
SAMPLE OUTPUT

```
TS01 6.000 29.559
TS02 174.413 19871.147
TS03 3622.000 10613591.698
TS04 7454.156 35964379.233
```

Problem H : Billard Paths 2

Input from file : `h.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

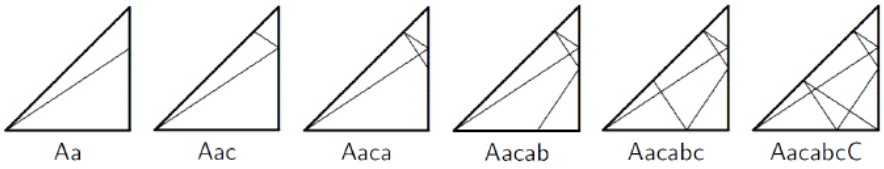
Let a point move on a frictionless plane bounded by a right isosceles triangle. If the point hits a vertex, then it stops. If it hits a side, then it changes its direction of motion such that the angle of reflection is equal to the angle of incidence. The path that the point follows is called a billiard path.



Let the triangle have vertices A, B, and C and sides a, b, and c as shown. Let a path start at vertex A with an initial slope $0 < m < 1$ chosen such that the path eventually ends at either vertex B or vertex C.

Let the path be described by a string that starts with (an uppercase) "`A`," continues with the lowercase letters describing the sides that the path hits in the order it hits them, and ends with the uppercase letter describing the vertex where the path ends.

For example, an initial slope of $m = 2/3$ corresponds to the billiard path described by the string "`AacabcC`." The figures below illustrate why this is so.



INPUT FORMAT

The input is a list of integers separated by single spaces. The first integer is the number of paths p. The remaining integers are numerator-denominator pairs, n and d, that describe the initial slope of the paths. (For example, two paths with initial slopes $m_1 = 2/3$ and $m_2 = 1/2$ would be described by the input "`2 2 3 1 2`." The slopes are not necessarily given in lowest terms so the input "`2 4 6 3 6`" describes the same paths as the input "`2 2 3 1 2`.")

OUTPUT FORMAT

The output is a list of strings, with each string on a separate line. Each string describes each path in the input (as specified above) with the strings given in the same order as the corresponding paths.

CONSTRAINTS

$0 < p < 100$
$0 < n < 100$
$0 < d < 100$
$n < d$

SAMPLE INPUT

```
9 1 6 2 6 3 6 4 6 5 6 1 5 2 5 3 5 4 5
```

SAMPLE OUTPUT

```
AacbcacbcacC
AacbB
AacC
AacabcC
AacabcbacabcbacC
AacbcacbB
AacbacacbcC
AacabcabcbB
AacabcbacabcC
```

Problem I : Counting Claws

Input from file : `i.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

In a network of N nodes $\{n_0, n_1, \ldots, n_{N-1}\}$, a pair of nodes n_i and n_j are connected to each other if there does exist a link (i, j), for all possible pairs i and j. A quadruplet of nodes n_1, n_2, n_3, and n_4 form a claw with n_1 as the head if the links $(1, 2)$, $(1, 3)$, and $(1, 4)$ exist. This network is undirected so when a link (i, j) exists, so is the link (j, i). The problem is to count the number of such claws given a network whose set of N nodes and set of M links are known. The network in Figure 1 below has six nodes and four claws: (n_1, n_0, n_2, n_3), (n_2, n_1, n_4, n_5), (n_3, n_0, n_1, n_4), and (n_4, n_2, n_3, n_5), with the first node in the quadruplets as the respective head nodes of the claws for easy visualization.

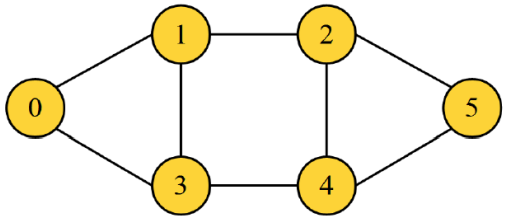


Figure 1: A network of $N = 6$ nodes (represented in this visualization as colored circles numbered from 0 to 5) and their links with each other (represented as lines connected to circle).

INPUT FORMAT

The input consists of several cases to be processed. The first line is a positive integer T, indicating the number of cases. The data for each of the T cases has the following specifications:

1. The first line of each case is a positive integer N which indicates the number of nodes in the network. The nodes are indexed as $\{0, 1, \ldots, N - 1\}$.

2. The next line contains a positive integer M which means the number of links. Note that $M \leq N(N - 1)$.

3. In the $M/2$ lines that follow, each contains a pair of non-negative integers i and j, where each pair is separated by at least one whitespace. Note that $i \neq j$, both must not be greater than $N - 1$, and if (i, j) is explicitly defined in the file, then (j, i) is implicitly defined as well.

OUTPUT FORMAT

For each case in the input, print the corresponding number of claws in one line.

CONSTRAINTS

$0 < T < 6$

SAMPLE INPUT

```
2
5
12
0 1
1 2
1 3
1 4
2 4
3 4
4
12
0 1
0 2
0 3
1 2
1 3
2 3
```

SAMPLE OUTPUT

```
5
4
```

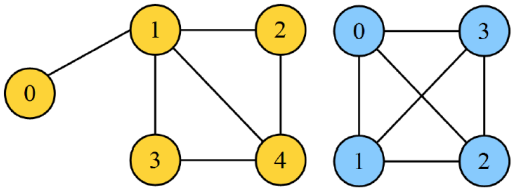


Figure 2: The visualization of the networks in the example input file. The first case has five nodes and five claws while the second, a completelyconnected network, has four nodes and four claws.

Problem J : Palindromes

Input from file : `j.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

Given a positive integer N, find the biggest number $P \leq N$ such that P is a palindrome. That is, P has the same value when read left-to-right as when read right-to-left. For example, when $N = 123456789$, then $P = 123454321$ is the biggest palindrome that is less than or equal to N. For another example, when $N = 234582345$, then $P = 234575432$ is the biggest palindrome that is less than or equal to N.

INPUT FORMAT

The input consists of several data sets. Each data set consists of one line of input data; the line starts with the data set ID (at most six alphanumeric characters) followed by the value of N, the two separated by a space.

The next data set will immediately follow the previous data set.

A data set ID of "ZZ" will signify the end of input data.

OUTPUT FORMAT

For each data set, print the data set ID, and the values of P and N, separated by a space.

CONSTRAINTS

$0 < N \leq 2,147,483,647$

SAMPLE INPUT

```
PAL01 1234567890
PAL02 9876543210
PAL03 12345678900
PAL04 98765432100
PAL05 12345012345
PAL06 98765098765
ZZ
```

SAMPLE OUTPUT

```
PAL01 1234554321 1234567890
PAL02 9876446789 9876543210
PAL03 12345654321 12345678900
PAL04 98765356789 98765432100
PAL05 12344944321 12345012345
PAL06 98765056789 98765098765
```

Problem K : Real or Imaginary?

Input from file : `k.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

Given a **quadratic equation** $ax^2 + bx + c = 0$, the **discriminant** of this equation is given by $D = b^2 - 4ac$.

Define $y = \frac{k}{\sqrt{D}}$ where k is a constant real number (either negative, zero or positive). We can classify y to be any of the following:

`A`: imaginary, if $D < 0$

`B`: indeterminate, if $k = 0$ and $D = 0$

`C`: undefined, if $k \neq 0$ and $D = 0$

`D`: negative real number, if $k < 0$ and $D > 0$

`E`: positive real number, if $k > 0$ and $D > 0$

`F`: zero, if $k = 0$ and $D > 0$

Your task is to determine the classification of y given the values of a, b, c, and k.

INPUT FORMAT

The input contains an arbitrary number of input lines. Each input line contains 4 representing a, b, c, and k (in that order) separated by a space.

OUTPUT FORMAT

For each input line, the output is the case number (i.e., the first input line is 1, second is 2, etc.) and the classification (either `A`, `B`, `C`, `D`, `E`, or `F`) separated by a space.

CONSTRAINTS

The maximum number of input lines is 1,000.

SAMPLE INPUT

```
0 0 0 0
1 2 3 4
1 5 1 1
1 10 2 -5
0 0 0 1
```

SAMPLE OUTPUT

```
1 B
2 A
3 E
4 D
5 C
```

Problem L : I love U

Input from file : `l.in`
Output to console: `stdout` (in C), `cout` (in C++), `System.out` (in Java)
Execution time limit: 2 seconds

Maria is a young girl with a fascination for the capital letter U.Whenever she reads a sentence and sees the capital letter U, Maria makes a mental count of it.

Given a string of characters, your task is to create a computer program that will help Maria to count the occurrence of the capital letter U.

INPUT FORMAT

The input contains an arbitrary number of input lines. Each input line is a case and contains a string of characters that may include whitespaces.

OUTPUT FORMAT

For each input line, the output is the case number (i.e., the first input line is 1, second is 2, etc.) a space and the number of times the capital letter U occurred in the string.

CONSTRAINTS

The maximum number of characters in a line is 255 and the maximum number of lines in the input is 1,000.

SAMPLE INPUT

```
The quick brown fox jumps over the lazy dog.
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.
The qUick brown fox jumps over the lazy dog.
The qUick brown fox jUmps over the lazy dog.
The q$uick brown fox j@mps over the lazy dog.
```

SAMPLE OUTPUT

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