# Problems and Solutions
# **IOPC 2012**

by

Raziman T.V. and Utkarsh Lath

Problem Code : IOPC1200

## Hardware upgrade[1]

It is after years that Dystopian Institute of Technology has got funds for upgrading their computers. However, the people in charge of the upgrade have decided to make this a chance to fill their own pockets rather.

There are $N$ computers in the DIT network, numbered 0 to $N-1$. Computer 0 is the main server which connects the network to the outer world. Some pairs of the computers are directly connected in the network. For other pairs of computers to communicate with each other, they have to do it via some other computer. For example, if there are only three computers in the network and the only direct connections are $0-1$ and $0-2$, then 1 communicates with 2 using 0 as intermediate. There is no limit on the number of intermediate computers to be used for communication between a pair. Now, since the DIT network was built on the principle of minimum expenditure, the $N$ computers have been made pairwise connected by using the minimum number of direct connections - ie, $N-1$.

The upgrade contractor has decided to not do the work properly but pocket the entire funds instead. However, to show that he has done something, he will rearrange the computers. For example he could move the computer number 3 to where 2 was earlier, 2 to 1, 1 to 3, 4 to 5 and 5 to 4. The computer number 0 is special and cannot be moved. Direct connections between computers depend on the locations though. Hence if there was a direct connection earlier between 3 and 5, it will now be between 1 and 4 (since they have been placed at the locations 3 and 5 were at earlier). However, he has noticed something strange : the new direct connection between computers i and j works if and only if there was a direct connection between i and j earlier. Hence he wants to rearrange the computers in such a way that the pairs of computers which are connected directly now are the same pairs which were connected earlier.

Given the computer network, count the number of ways the computers can be rearranged satisfying the conditions.

## Input

The first line of the input contains T, the number of test cases ($1 \leq T \leq 10$). Following this are the descriptions of the test cases.
Each description starts with a line containing the integer $N$, the number of computers in the network ($1 \leq N \leq 10000$). This is followed by $N-1$ lines, each containing a pair of integers $i$ and $j$, denoting that computer $i$ is directly connected to computer $j$. It is assured that these $N-1$ direct connections will make every pair of computers connected (via intermediates if necessary).

## Output

For each test case, output the number of rearrangements of the computers. A rearrangement $(p_0, p_1, p_2 ... p_{N-1})$ of $(0, 1, 2 ... N-1)$ is valid if the following conditions are satisfied:

- $p_0 = 0$
- If there is a direct connection between $i$ and $j$, there should also be a direct connection between $p_i$ and $p_j$

Notice that $(0, 1, 2 ... N-1)$ is considered a valid rearrangement of itself. Also, since the answer could be very big, output it modulo 1000000007.

[1]Problem formulated by Raziman T V

## Example

**Input:**

1
6
0 1
0 2
0 3
3 4
3 5

**Output:**

4

## Solution

The problem here was just to count the number of isomorphisms of a rooted tree. A tree with $n$ nodes has $n$ rooted subtrees. One needs to find out which rooted subtrees are isomorphic to each other. Suppose that we are able to assign a unique label to all rooted subtrees which are isomorphic to each other, then the problem can be solved. We can do this recursively. Keep a map from a vector of integers to integers - this is to store the unique labels of subtrees. Start with assigning the label 0 to the leaf subtree. While at a node $v$, make a vector of the labels of its children and sort the labels. If a label $l_1$ appears $n_1$ times, $l_2$ appears $n_2$ times and so on, the number of isomorphisms that can be made by just relabeling the children of the $v$ is $n(v) = n_1!n_2!...$. Now check if this sorted vector exists in the map. If so, read the label for this vertex from the map. Otherwise, add it to the map with a new label. Do this for all vertices, in reverse dfs order to make sure that children of a node are processed before the node. Multiplying the values of $n(v)$ for all vertices gives the number of isomorphisms.

Problem Code : IOPC1201

# Rubik's cube[2]

The Rubik's cube is perhaps the world's most popular intellectual toy. More than just the joy of solving, there is a lot of mathematics to it too.

Consider a solved Rubik's cube. The six faces of the cube are named FRONT, BACK, UP, DOWN, LEFT and RIGHT respectively. An elementary move of the Rubik's cube is rotating a face by $90^\circ$ clockwise, $90^\circ$ anticlockwise or $180^\circ$ about an axis from the centre of the face to the centre of the cube. Any valid state of the Rubik's cube can be reached by applying these elementary operations one after the other.

An elementary move is denoted in the following fashion. If a given face is rotated by $90^\circ$ clockwise about the axis passing from the centre of the face to the centre of the cube, the move is denoted by the first letter of the name of the face. If the rotation is anticlockwise by $90^\circ$, the letter is followed by an apostrophe ('). If the rotation is by 180 degrees, the letter is followed by a 2.

For example, a $90^\circ$ clockwise rotation of the right face is denoted by *R,* a $90^\circ$ anticlockwise rotation of the back face by *B'*, and a $180^\circ$ rotation of the top face by *U2*. The elementary moves are explained with the help of animations in the *Face turns* section of this page. The elementary moves can be combined to get a compound move of the cube. For example, *URF2* denotes rotating the top face by $90^\circ$ clockwise, then the right face by $90^\circ$ clockwise followed by the front face by $180^\circ$.

In this problem you will be given a string describing a sequence of elementary moves on a solved Rubik's cube. Your task is to find out the number of times the sequence should be applied repeatedly to the cube to get back the original cube.

## Input

The first line of input gives T, the number of test cases ($1 \leq T \leq 1000$). Following this are T lines, each containing a string denoting a sequence of moves on the Rubik's cube. The string will contain only the characters *U*, *D*, *L*, *R*, *F*, *B*, *'* and *2*, and will be at most 1000 characters long. It is assured to be a valid sequence of elementary moves.

## Output

For each test case, output the minimum number of times ($\geq 1$) the move sequence must be applied to a solved Rubik's cube to get back the solved cube again.

[2]Problem formulated by Raziman T V

## Example

**Input:**

4
R
RR2
RU
R2R'R'

**Output:**

4
4
105
1

## Solution

A move of the Rubik's cube is just a permutation of its cubies. This permutation can be thought of being over the 48 movable coloured unit squares or the 20 movable coloured unit cubes (8 corner pieces and 12 edge pieces) of the cube. Either way, find out the permutation that an elementary move does to the cube. Compose these elementary permutations to find out the permutation corresponding to the compound move. Now the order of the permutation can be found using cycle decomposition.

Problem Code : IOPC1202

## Quadrilaterals[3]

You are given the coordinates of the vertices of a square in the 2-d plane. All vertex coordinates will be integers. Consider all convex quadrilaterals which also have integer coordinates for their vertices such that the midpoints of their edges are the vertices of the original square. Find the sum of areas of all such quadrilaterals.

## Input

The first line of input gives $T$, the number of test cases ($1 \leq T \leq 25000$). Following this are the descriptions of the individual test cases.

Each testcase consists of four lines, each line containing two space separated integers - the $x$ and $y$ coordinates of a distinct vertex of the square. The coordinate limits are $-1000000000 \leq x_i, y_i \leq 1000000000$. It is assured that the four vertices will correspond to a square.

## Output

For each testcase output the total area of quadrilaterals with the given property. Since the answer could be very large, give the answer modulo 100000007.

## Example

**Input:**

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

**Output:**

```
0
4
```

[3]Problem formulated by Raziman T V

## Solution

If one of the vertices of the quadrilateral is given, the other vertices can be easily found out. Let $S'$ be the square generated by translating the original square $S$ parallel to one of its sides by a distance equal to the side length. It can be proved that any convex, non-degenerate quadrilateral with midpoints of the edges as vertices of $S$ will have exactly one of its vertices lying completely inside $S'$. So it is enough to sum the areas of such quadrilaterals with one vertex in $S'$. But all the quadrilaterals have the same area - equal to twice the area of $S$. Hence we count the number of lattice points in $S'$ (using Pick's theorem or otherwise) to get the number of quadrilaterals and multiply it by twice the area of $S$ to find the answer.

Problem Code : IOPC1203

## Crazy texting[4]

You must be familiar with the use of numeric keys to enter alphabets in mobile phones. A single numeric key when pressed gives a character. When pressed again, it changes to another and so on. Once the set of characters mapped to the key is exhausted, the key wraps around to the original character.

The key mapping in this problem is the T9 mapping, restricted to lowercase english characters. The characters corresponding to the individual keys are:

- 2: a,b,c
- 3: d,e,f
- 4: g,h,i
- 5: j,k,l
- 6: m,n,o
- 7: p,q,r,s
- 8: t,u,v
- 9: w,x,y,z

For example, if you press the key 2 once, it prints the character '*a*'. On pressing it again, it becomes '*b*', then '*c*', then back to '*a*' and so on.

Consider a string made of lowercase letters only. To enter this string into a mobile phone, a certain key sequence has to be entered with sufficient gaps in between. Suppose that the key sequence entered is correct but the gaps between keypresses are made arbitrarily. This can result in very different strings being printed.

For example, let the string to be input be "*mod*". The key sequence corresponding to this is *6_6663* where '_' denotes a gap between the keypresses. Suppose the keys are pressed in the same order, but with gaps between keypresses arbitrary. This can result in 8 different strings: "*mod*", "*nnd*", "*omd*", "*mmmmd*", "*mnmd*", "*mmnd*", "*nmmd*" and "*md*".

Given an input string, find the number of possible strings printed by the minimal key sequence corresponding to it.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$). Following this are $T$ lines, each containing the input string. The string will consist only of lowercase letters and will have a maximum length of 100000.

## Output

For each string, output the number of T9 strings corresponding to its minimal key sequence. Since the answer can be very large, output it modulo 100000007.

[4]Problem formulated by Raziman T V

## Example

**Input:**

2
mod
iopc

**Output:**

8
64

## Solution

In this problem we could split the input string into contiguous segments using the same key and find the answers for those segments separately, and multiply them to get the final answer. So suppose the string under consideration can be got by pressing the key "2" $n$ times (The key has a wraparound length of 3, a wraparound of 4 can be dealt with similarly). Define $g(n)$ as the number of distinct strings which can be got by pressing "2" $n$ times. Now, this string might have had anything between 0 to $\frac{n}{3}$ wraparounds. Suppose $f(n)$ is the number of distinct strings which can be made by pressing the key "2" $n$ times, with wraps not allowed. Then $g(n) = f(n) + f(n-3) + f(n-6) \ldots$. Also $f(n) = f(n-1) + f(n-2) + f(n-3)$. Using this and a similar recurrence for the 4-wrap keys, we can solve the problem.

Problem Code : IOPC1204

## A function over factors[5]

A function f is defined over natural numbers as:

$$f(N) = \sum_{d|N} d\mu(d)$$

Here the summation is over all positive integers $d$ which are factors of $N$.

$\mu(n)$ is the Möbius function defined in the following way: If there exists a prime $p$ such that $p^2$ is a factor of $n$, then $\mu(n) = 0$. Otherwise, if $n$ has an odd number of prime factors, $\mu(n) = -1$. If not, $\mu(n) = 1$. Thus the first few values for $\mu(n)$ (starting from 1) are $1, -1, -1, 0, -1, 1, -1, 0...$

Given an integer X ($0 \leq X \leq 10^{12}$), find the smallest natural number N such that $|f(N)| > X$.

### Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 1000$). Following this are $T$ lines, each containing an integer $X$ ($0 \leq X \leq 10^{12}$) corresponding to the test case.

### Output

For each test case in the input, output the smallest natural number $N$ such that $|f(N)| > X$.

### Example

**Input:**

2
1
2

**Output:**

3
5

### Solution

Suppose $n = p_1^{q_1} p_2^{q_2} ... p_r^{q_r}$ where the $p_i$s are distinct primes. Then it can be seen that $f(n) = (1-p_1)(1-p_2)...(1-p_r)$. Using the fact that gaps between primes is less than $\sqrt{n}$ for primes around $n$, we can prove that if $n$ is composite, there exists a prime $p < n$ such that $|f(p)| > |f(n)|$. Also, $f(p) = 1 - p$. Hence to find the smallest number with $|f(n)| > x$, it is enough to find the smallest prime $p \geq x+2$. This can be done by eliminating most of the candidates using sieving and using Miller-Rabin for the rest.

[5] Problem formulated by Utkarsh Lath

Problem Code : IOPC1205

## The magical escape[6]

The nasty king of Dystopia has captured a group of travellers from Utopia. The king wants to toy with them, so he makes them play a little "game":

The king will place everyone in solitary confinement. He has placed a bulb in a separate room which is controlled by a switch. Initially the bulb is off. Each day he picks a prisoner at random and takes him to the room. He can see whether the bulb is on or not, and choose whether to toggle the state of the bulb. This continues day after day. Now the king says that, once a prisoner realises that all of them have visited the special room at least once, he can convey it to the king and he will set everyone free. However, if someone makes such a claim wrongly, everyone will be executed. Before locking everyone up, the king would allow all prisoners to get together and decide a strategy.

This would usually be a very diabolic game to play - However, among the prisoners there were some magicians who could communicate with each other telepathically. So they come up with this strategy. All magicians start counting from 0. Every time a non-magician who has never toggled the state of the bulb before enters the room and finds the lamp off, he turns it on. However if he finds it on already, or if he has toggled the state of the bulb before, he leaves it as is. Whenever a magician enters the room, he does the following:

- If he is entering the room for the first time, he increases his counter by 1 and telepathically communicates this to all other magicians who update their counter as well.
- If he finds the lamp on, whether or not he is a first timer, he turns it off and all magicians increase their counter by one.

Finally when the magicians' counter reaches the total number of prisoners, they declare that everyone has been in the room at least once.

Find the expected number of days that pass before this declaration occurs.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 1000$). Following these are T lines corresponding to the test cases.
Each line will contain two space separated integers $N$ and $M$, the total number of prisoners and the number of magicians among them respectively ($1 \leq N \leq 10^9$, $1 \leq M \leq \min(100, N)$).

## Output

For each test case, output the expected number of days before the magicians declare that all prisoners have entered the room at least once. The output has to be formatted in the following fashion: #.######E+## That is, one nonzero digit before the decimal point, six digits after the decimal point and two digits for the exponent. See the sample test case for a formatting example.

[6] Problem formulated by Utkarsh Lath

## Example

**Input:**

2
100 10
1 1

**Output:**

1.408258E+03
1.000000E+00

## Solution

Consider the scenario where the magicians had to only find out when all non-magicians (rather than everyone) had visited the room. If there are $n$ non-magicians and $m$ magicians ($N = m+n$), the expected time can be found to be $g(n,m) = \frac{Nn}{m} + NH_n$. Now consider the actual case where they have to find out when everyone has visited the room. Define the expected time remaining with $i$ non-magicians out of $n$ and $j$ magicians out of $m$ yet to enter the room and the current state of the bulb off as $f(n,m,i,j)$. Let $N = m+n$, $i' = n-i$ and $j' = m-j$. Then

$$f(n,m,i,j) = \frac{1}{i+j}\left[\frac{N(m+i)}{m} + jf(n,m,i,j-1) + \frac{ij'}{m}f(n,m,i-1,j) + \frac{ij}{m}f(n,m,i-1,j-1)\right]$$

This recurrence can be used to find the answer $f(n,m,n,m)$ in $O(nm)$ time. However, that is too slow for the constraints in the problem. Hence we make the following observation. With $n$ large enough, the expected time taken for all non-magicians to be declared as having visited the cell goes as $n^2$. The probability that a certain magician has not visited the cell at all in these many number of steps is truly small, and goes like $e^{-N}$. Hence if one has $n > 1500$, say, then for $m < 100$, there is very little difference between $g(n,m)$ and $f(n,m,n,m)$ - less than the required precision in the problem. Hence one can use the second formula for small values of $n$ and the first formula for large values. Euler formula for approximating the partial sums of the harmonic series can be used to find $H_n$ with the required precision.

Problem Code : IOPC1206

## Fair bases[7]

Consider integers $N$ and $K$ such that $2 \le K \le N$. Write all the numbers $0 \le i < N$ in base $K$, adding leading zeros where necessary so that all the numbers are of equal length in base $K$. The score of an integer $i$ ($0 \le i < N$) in the list is defined in the following fashion : Consider the first digit of $i$ in base $K$. Count the total number of times this digit occurs as first digit of some integer in the list. This is the score of the first digit of $i$. The number of times the second digit of $i$ appears as the second digit of some integer in the list is the score of the second digit of $i$, and so on. The sum of scores of all digits of $i$ is the score of $i$.

As an example, suppose $N = 4$ and $K = 3$. Then the numbers in the list are 00,01,02 and 10. Let us find the score of $i = 00$. The first digit of $i$ appears as the first digit thrice (00,01,02) and the second digit of $i$ appears as second digit twice (00,10). Thus the score of 00 is $3 + 2 = 5$.

An integer K ($2 \le K \le N$) is called a fair base of $N$ if the scores of all i ($0 \le i < N$) are equal for base $K$. The number of fair bases in the range $2 \le K \le N$ is termed the fairness factor of the integer $N$.

Given integers a and b ($2 \le a \le b \le 10^{12}$), find the sum of fairness factors of all $i$ such that $a \le i \le b$.

### Input

The first line of the input consists of $T$, the number of testcases ($1 \le T \le 20$). Following this are $T$ lines, each containing two space separated integers a and b ($2 \le a \le b \le 10^{12}$).

### Output

For each $(a, b)$ pair in the input, output the sum of fairness factors of $i$ in the range $a \le i \le b$.

### Example

**Input:**

2
4 8
20 30

**Output:**

9
27

[7]Problem formulated by Raziman T V

## Solution

If $K$ is a fair base for $N$, then $N = jK^l$ for some $j < K$ and $l > 0$. ie, $N$ can be written as $j00...0$ in base $K$. So we need to find the total of such $K$s from $n = 1$ to $N$. First see which all numbers can be written as 10 in some base. This is possible for all numbers except 1 - a total of $N-1$. Similarly $\frac{N}{2} - 2$ numbers can be written as 20 and so on. In general, $\sqrt[r]{\frac{N}{j}} - j$ numbers can be written as $jK^r$ for some $K > j$ if the quantity is non-negative. Hence what we want is the sum

$$\sum_{r=1}^{log_2 N} \sum_{j=1}^{N^{\frac{1}{r+1}}} \left[ \sqrt[r]{\frac{N}{j}} - j \right]$$

Some optimisation for finding the quantities and summing is needed for the solution to pass within the time limit. For example, one could precompute all perfect powers in an array and use binary search and incremental search smartly on the array to find the roots.

Problem Code : IOPC1207

## GM Plants[8]

The latest attraction for Techkriti 2112 is is a huge display of genetically modified plants. The arrangement consists of a cuboidal box of size $N_x \times N_y \times N_z$ made of unit cubes. Each unit cube is identified using $(x, y, z)$ coordinates - $x$ ranges from 0 to $N_x - 1$ and so on - and contains a plant genetically modified to show fluorescence. The natural colour of the plants is green. However, on exposing a plant to laser light, it changes colour to red. What is more interesting is that on exposing a red plant to laser light again, it changes back to green and this continues.

The organisers have realised that they can use the display to make many coloured patterns. They have with them a laser light sheet which they can place along an axis and move in one direction, exposing many plants to light at once. For example, if the plane of the light sheet is kept as the $y$ axis and it is moved from $a$ to $b$, every plant with the $y$ coordinate between $a$ and $b$ inclusive will turn from green to red or red to green. Every time the laser is operated, it is only the plants with one specific coordinate in a certain range which are affected.

You are told that initially all plants were green. Given the sequence of exposing plants to laser light, your task is to find the number of red coloured plants in certain cuboidal subregions of the display

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$). Following this are the descriptions of the T test cases.
The description of each test case starts with a line containing four space separated integers : $N_x$, $N_y$, $N_z$ and $Q$ ($1 \leq N_x, N_y, N_z \leq 100000$, $Q \leq 5000$). The first three are the extents of the display in the three dimensions while $Q$ is the number of queries which are to follow. Following this are $Q$ lines, each describing a query. A query will be of one of the following forms :

- 0 $i$ $j$ : expose all plants with $x$ coordinates $i \leq x \leq j$ to laser light
- 1 $i$ $j$ : expose all plants with $y$ coordinates $i \leq y \leq j$ to laser light
- 2 $i$ $j$ : expose all plants with $z$ coordinates $i \leq z \leq j$ to laser light
- 3 $x_1$ $y_1$ $z_1$ $x_2$ $y_2$ $z_2$ : Report the number of red plants in the cuboidal region with $(x_1, y_1, z_1)$ and $(x_2, y_2, z_2)$ as diagonally opposite cells - ie, all red plants with $x_1 \leq x \leq x_2$ ,$y_1 \leq y \leq y_2$ and $z_1 \leq z \leq z_2$

All individual coordinates will be valid - ie, every $x$ coordinate will be such that $0 \leq x < N_x$ and so on. Also, $1 \leq j$, $x_1 \leq x_2$ and so on.

## Output

For every query of the form 3 $x_1$ $y_1$ $z_1$ $x_2$ $y_2$ $z_2$ in the input, output the number of red plants with coordinates constrained by $x_1 \leq x \leq x_2$ , $y_1 \leq y \leq y_2$ and $z_1 \leq z \leq z_2$

[8] Problem formulated by Raziman T V

## Example

**Input:**

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

**Output:**

12
4

## Solution

Say we need to find the count in the range $(i_1, j_i, k_1 - i_2, j_2, k_2)$. Since the cells are flipped based on only one coordinate, we can split this problem into three $1-d$ problems of finding the red ranges in $(i_1 - i_2)$, $(j_1 - j_2)$ and $(k_1 - k_2)$. Suppose the red and green lengths in these ranges are $r_1$, $g_1$, $r_2$, $g_2$, $r_3$, $g_3$ respectively. Then the answer we need is just $r_1r_2r_3 + r_1g_2g_3 + g_1r_2g_3 + g_1g_2r_3$. The individual range sums can be found by doing the update and query operations using a segment tree. The values to be stored at each node of the segment tree are the number of red cells in the subtree and whether the interval has been totally flipped or not. While updating, only toggle the flip bits of the required intervals and propagate the sums up. While querying, also count the number of flip bits encountered while traversal from the root.

Problem Code : IOPC1208

## IOPC Transformation[9]

Given natural numbers $N$ and $Q$, an $N$-vector is defined as the ordered $N$-tuplet of integers $(x_0, x_1, x_2 \ldots .x_{N-1})$ such that $0 \leq x_i < Q$ for all $i$. The parent of coordinate $x_i$ is defined as $x_{(i-1) mod N}$ and the child of $x_i$ is defined as $x_{(i+1) mod N}$

The IOPC transformation converts and $N$-vector $x$ into an $N$-vector $y$ with the coordinates given by :

- If parent($x_i$) $\leq$ child($x_i$) then $y_i$ = (parent($x_i$) + child($x_i$))/2
- If parent($x_i$) > child($x_i$) then $y_i$ = (parent($x_i$) + child($x_i$) + Q)/2 mod Q

Note that the division above is integer division - fractions are always rounded down.

As an example, let $N = 4$, $Q = 5$. Let the $N$-vector $x$ be $(2, 0, 3, 1)$. Then the transformed $N$-vector $y$ is $(3, 2, 0, 0)$.

An $N$-vector $x$ is said to be invariant under the IOPC transformation if $y_i = x_i$ for all values of $i$. Given $N$ and $Q$, find the number of invariant $N$-vectors.

### Input

The first line of the input consists of $T$, the number of testcases $(1 \leq T \leq 10^4)$.Following this are $T$ lines, each consisting of two space separated integers $N$ and $Q$ $(1 \leq N, Q \leq 10^9)$.

### Output

For each test case, output the number of invariant $N$-vectors. Since the answer could be very large, output it modulo 100000007.

### Example

**Input:**

3
2 7
4 5
6 3

**Output:**

7
5
6

[9]Problem formulated by Raziman T V. Thanks to Prof. A.K. Lal for suggesting the source.

## Solution

This problem was based on *The solution to a cyclic relaxation problem* by Edgar Dijkstra (EWD386). The point to note is that the N points can be thought of being placed around a circle of Q points - the transformation then just finds the clockwise midpoint of $x_{i-1}$ and $x_{i+1}$. For the vector to be invariant, we need $x_i$ to lie at this clockwise midpoint. Due to the cyclic symmetry, let $x_0 = 0$. Then consider $d_i = x_{i+1} - x_i$. Here the subtraction is clockwise subtraction. Since we need each point to be at the midpoint of its parent and child and that we should come back to the origin after moving through the $N$ points, we have that all $d_i$ s are equal and that the sum of $d_i$ s is a multiple of $Q$. Also, since the distances have to be clockwise we need $d_i < \frac{Q}{2}$. So what we want is just the number of solutions in this range of $Nd = mQ$. If $g = \gcd(Q,N)$ this is just $1 + \frac{\lfloor \frac{Q-1}{2} \rfloor}{Q/g}$. Multiply this with $Q$ to get the answer.

Problem Code : IOPC1209

## Enumeration again[10]

Those of you who participated in IOPC last time might remember the *Enumeration of rationals* problem. Here is a variant of the same:

We enumerate the rationals in $(0,1)$ in the following fashion. First, every rational is expressed in the lowest terms : ie, as $p/q$ where $p$ and $q$ are positive integers with no common factor other than one. Then we sort the fractions in the ascending order of $p+q$. In case of a tie, the smaller fraction comes first.

The first few terms in this enumeration are $1/2$, $1/3$, $1/4$, $2/3$, $1/5$, $1/6$, $2/5$...

Given an integer $N$, find the position of $1/N$ in this enumeration.

### Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 25$).Following this are $T$ lines, each containing an integer $N$ ($2 \leq N \leq 10^9$).

### Output

For each $N$ in the input, output the position of $1/N$ in the enumeration.

### Example

**Input:**

```
4
2
3
4
5
```

**Output:**

```
1
2
3
5
```

[10]Problem formulated by Utkarsh Lath

## Solution

What we need to find is just $f(N) = \sum_{i=1}^{N} \phi(i)$ where $\phi$ is the Euler's totient function. This can be evaluated in the following way. Consider all numbers such that $1 \le a \le b \le N$. There are $\frac{N(N+1)}{2}$ such numbers. Now count them separately based on $\gcd(a,b)$. There are $f(N)$ pairs such that $\gcd(a,b) = 1$, $f(\frac{N}{2})$ pairs such that $\gcd(a,b) = 2$ and so on. Hence:

$$f(N) = \frac{N(N+1)}{2} - f(\frac{N}{2}) - f(\frac{N}{3}) \ldots f(\frac{N}{N})$$

If we knew all the values of $f(i)$ from $i = 1$ to $N-1$, we can find $f(N)$ in $O(\sqrt{N})$ time by splitting the sum into $i < \sqrt{N}$ and $i > \sqrt{N}$. But since that would be costly, we can find $f(i)$ for $i$ upto some $M$ by sieving as precomputation and finding $f(\frac{N}{i})$ for $\frac{N}{i} > M$ using the above relation.

Problem Code : IOPC1210

## An unfortunate incident[11]

Terrorists from Dystopia had always been targeting Utopian civilians, but this is a new low that they have reached.

A psychotic terrorist, disguised as a teacher, has taken some kindergarten students hostage. She first made every student write the names of the other students (s)he likes on a piece of paper and collected these "like lists". Now she plans to do the following:

She will select a set of students at random from the lot. Then she will look at the "like lists" collected from the entire class. If a like list has the name of at least one selected student, it is considered a favourable like list. If the number of favourable like lists is greater than or equal to the number of students selected, she sends them back and picks another set of students. However, if the number of favourable like lists is less than the number of students selected, she plans to kill the selected students.

You will be given the like lists of all the students. Your task is find out whether the class is safe or not. The class is said to be not safe if the terrorist can pick some set of students such that the total number of favourable like lists for them is less than the number of selected students, resulting in their execution.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$).Following this are the descriptions of the test cases.

The description of each test case begins with a line containing the integer $N$, the number of students held hostage ($N \leq 500$). The students are numbered from 0 to $N-1$. Following this are $N$ lines describing the like lists : The first line after $N$ describes the like list of student 0, the second line the like list of student 1 and so on. Each like list contains some space separated integers. The first integer is $K$, the number of students the student likes ($0 \leq K \leq \min(50, N-1)$). Following this are the identities of the students (s)he likes. The identities of the students will be between 0 and $N-1$. You are assured that no number will appear more than once in the list, and that no student will have his own number in the list.

## Output

For each test case, output *safe* if all the children are safe and *not safe* otherwise, according to the definition in the problem statement.

[11] Problem formulated by Raziman T V and Utkarsh Lath

## Example

### Input:

2
3
2 1 2
2 2 0
2 0 1
3
2 1 2
1 0
1 0

### Output:

safe
not safe

## Solution

This problem can be solved by a direct application of the Hall's Marriage theorem : A perfect matching appears in a bipartite graph if and only if the neighbourhood of any subset of either of the two vertex sets has cardinality greater than or equal to the cardinaity of the subset itself. Make a bipartite graph between students and like lists, and see whether there is a perfect matching in the graph using flows. The students are safe if and only if a perfect matching exists.

Problem Code : IOPC1211

## The numbers game[12]

There is a new prodigy in town and he has challenged people to a game. They have to give him an integer $N$ and he will immediately respond with a number which has more than $N$ factors.

What the prodigy really does is to do some complex calculations in his head to find out the smallest integer with more than $N$ factors. However, he has a weakness - he knows to do those calculation only on numbers up to $10^{19}$, so if the answer to be reported is more than $10^{19}$, he will not be able to come up with his solution and will lose the game.

Given the integer the people give to the prodigy, you need to report whether he wins the game or not. If he wins, also output his answer.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 25$). Following this are $T$ lines, each containing an integer $N$ ($1 \leq N \leq 10^9$) - the number which is given to the prodigy.

## Output

Output one line per test case. If the smallest integer $X$ with more than $N$ factors is bigger than $10^{19}$ so that the prodigy does not win, print *lose*. Otherwise print *win X*.

## Example

**Input:**

3
3
5
12345678

**Output:**

win 6
win 12
lose

## Solution

Every number which is a solution to this problem will have more factors than any number less than it. But these are exactly the highly composite numbers, and there are less than 200 of them in the given range. So the problem can be solved by hardcoding the highly composite numbers in the code.

[12]Problem formulated by Raziman T V and Utkarsh Lath

Problem Code : IOPC1212

## Utopia strikes back[13]

The terrorist strike on the kindergarten was universally condemned. The Utopians' patience had worn thin and they decided to wipe out the entire Dystopian terrorist base.

Utopian military plans to do this with a coordinated strike. They have $N$ military bases distributed all over the world. They also know the locations of the $M$ terrorist camps. They will choose $M$ out of their $N$ military bases as attack bases and each will send a missile to one terrorist camp, all at the same time. By this they hope that they can eliminate Dystopian terrorism completely without giving them a chance to retaliate.

You are given the locations of the military bases and the terrorist camps. Given that all missiles have the same maximum range, your task is to find out the minimum value of the missile range so that such a coordinated attack is possible. Of course, the minimum distance between any pair of points in this case is the geodesic distance on the spherical planet.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$). Following this are the descriptions of the T test cases.
The description of each test case starts with a line containing three space separated integers $N$, $M$ and $R$ ($1 \leq M \leq N \leq 100$; $1 \leq R \leq 10000$). $N$ and $M$ are the number of military bases and terrorist camps respectively while $R$ is the radius of the spherical planet. Following this are $N$ lines containing the locations of the military bases - the locations are specified using two space separated numbers and , the latitude and longitude of the location on the planet in degrees. ranges from $-90$ to 90 while ranges from $-180$ to 180. Both numbers have 6 digits after the decimal point. Following these lines are $M$ lines with the locations of the terrorist camps in the same fashion.

## Output

For each test case, output the minimum value of the missile range required so that the coordinated attack can be carried out. Format this number in the following fashion : #.######E*## - Here, # stands for a digit and * stands for a sign (+/-). The leading digit has to be non-zero.

[13] Problem formulated by Raziman T V

## Example

**Input:**

1
4 2 10
0.000000 0.000000
0.000000 90.000000
0.000000 180.000000
0.000000 -90.000000
90.000000 0.000000
-90.000000 0.000000

**Output:**

1.570796E+01

## Solution

The question to be answered is: Given pairwise distances for $M$ points of type $A$ and $N$ points of type $B$, what is the way to match each point of A with a point in B such that the maximum distance is minimised? Instead we try to answer this question: Given $M$ points of type $A$ and $N$ points of type $B$, is there a way to match each point of A with a point in B such that each of the matched distances is less than $d$? But this can easily be solved using bipartite matching. Then we can do a binary search on $d$ to find the minimum value.

Problem Code : IOPC1213

## Colouring graphs[14]

In this problem you will be given a simple acyclic graph and asked to perform update or query operations on the graph. Update operations will involve adding an edge to the graph or removing one. You are assured that adding the given edges will not introduce cycles in the graph. Your task is to answer the queries which ask you to find the number of ways the graph can be coloured using $K$ colours such that no two adjacent vertices have the same colour.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 100$). Following this are the descriptions of the test cases.
The description of each test case starts with a line containing three space separated integers $N$, $M$ and $Q$ ($1 \leq N \leq 100$, $0 \leq M < N$; $Q \leq 1000$). $N$ and $M$ are the number of vertices and edges in the graph respectively (both indexed from 0), while $Q$ is the total number of update and query operations you will be asked to perform. Following this are $M$ lines of the form $i$ $j$, denoting that there is an edge between $i$ and $j$. This is followed by $Q$ lines, each of which will be in one of the three forms:

- 0 $a$ $b$ : Add an edge between the existing vertices $a$ and $b$
- 1 $a$ $b$ : Remove the existing edge between the vertices $a$ and $b$
- $K$ ($K > 1$) : Report the number of colourings of the graph with $K$ colours such that no pair of adjacent vertices is coloured same

You are assured that the graph remains simple and acyclic at every stage.

## Output

For each query of type $K$ ($K > 1$), output the number of ways the graph can be coloured validly using $K$ colours. Since the answer could be very large, output it modulo 100000007.

[14] Problem formulated by Raziman T V

## Example

### Input:

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

### Output:

48
12
4

## Solution

A tree of $N$ vertices can be coloured using $K$ colours in $K(K-1)^{N-1}$ ways. Suppose there is a forest which has $c$ components. Then the number of ways of colouring it is $\prod_c K(K-1)^{N_c-1} = K^c(K-1)^{N-c}$. And $c$ can be found just from the total number of vertices and edges as $c = N - M$.

Problem Code : IOPC1214

## Celebrities of Pretendia[15]

The celebrities of Pretendia are just as snobbish as those anywhere else. If celebrity $x$ admires celebrity $y$, Then $y$ thinks too high of him/herself and does not admire $x$ back. Also, if $x$ admires $y$, $x$ also admires everyone that $y$ admires.

All celebrities of Pretendia are going to be present at the premiere of *Chapangg 2*. True to their snobbish nature, the celebrities want to make sure that they do not wear dress made by the same designer who makes dress for someone who admires them. Given who admires whom in Pretendia, find out the minimum number of designers required to make dresses for all celebrities.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$). Following this are the descriptions of the test cases.
Each starts with two space separated integers $N$ and $M$ ($1 \leq N \leq 1000$; $0 \leq M \leq 40000$). $N$ is the number of celebrities in Pretendia and celebrities are numbered from 0 to $N-1$. Following this are $M$ lines of the form $i$ $j$ ($0 \leq i, j < N$) and tells you that celebrity $i$ admires celebrity $j$. Taken together, these $M$ lines will satisfy the conditions in the problem statement.

## Output

For each test case, output the minimum number of designers required so that every celebrity can wear a dress made by someone who does not design dresses for any of his admirers.

## Example

**Input:**

```
1
5 8
0 2
1 2
0 4
1 3
2 3
0 3
1 4
2 4
```

**Output:**

```
4
12
4
```

[15] Problem formulated by Raziman T V and Utkarsh Lath

## Solution

Since the graph is assured to be the transitive closure of a DAG, we just need to find the length of the longest chain in the graph and that will be equal to the minimum number of colours need to colour its vertices. The length of the chain can be found easily using a depth first search.

Problem Code : IOPC1215

## No to corruption[16]

Dystopians have realised that corruption is the root cause for all their problems. They have decided to launch their mass protest against corruption by blocking one of the highways in the Dystopian capital. To make sure that the government notices their agitation, they want to make sure that the Prime Minister is also affected by the blocking. Their aim is to force the Prime Minister to choose an alternate route from the parliament to his house, which makes his journey longer

All roads in the capital are two-way. The road network is such that no more than one road connects a pair of junctions directly. Also, every junction in the capital is connected to every other junction by some sequence of roads. Given the road map of the capital, find out the minimum distance the Prime minister has to travel to reach home from the parliament if the anti-corruption movement blocks one of the roads.

## Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 10$). Following this are the descriptions of the test cases.
Each starts with a line containing three space separated integers $N$, $M$ and $Q$ ($1 \leq N \leq 1000$; $1 \leq M \leq 20000$; $Q \leq 10000$). $N$ is the number of junctions in the capital and are numbered from 0 to $N-1$ (0 is the location of the Parliament and $N-1$ is the location of the Prime Minister's home). $M$ is the number of roads in the capital (numbered from 0 to $M-1$) and $Q$, the number of queries you have to answer. Following this are $M$ lines, each containing three space separated integers $i$, $j$ and $d$ ($0 \leq i, j < N$; $1 \leq d \leq 10^5$), denoting that there is a road between junctions $i$ and $j$ having distance $d$. $Q$ lines follow this, describing the queries you need to answer. Each query will contain a single integer $m$, the index of the road that will be blocked ($0 \leq m < M$).

## Output

For each query $m$ in the input, output an integer - the minimum distance the Prime minister has to travel to reach home from the Parliament if the anti-corruption movement blocks only the road $m$. If blocking the road prevents the Prime Minister from reaching home, print *no route for corrupt ministers!!!*

[16]Problem formulated by Utkarsh Lath

## Example

**Input:**

1
4 4 2
0 1 2
1 2 2
0 2 3
2 3 4
2
3

**Output:**

8
no route for corrupt ministers!!!

## Solution

Consider the shortest path tree got from a Dijkstra at the source. When an edge on the shortest path from the source to the sink is removed, the tree is split into two components. The shortest path then from the source to sink will involve a path along the shortest path tree from source to some vertex in that component, an edge from that vertex to a vertex in the other component and the shortest path from that vertex to the sink. Shortest paths from the source and sink can be found using two Dijsktras. Sort all vertices by the closest vertex on the shortest path tree from it. Now traverse from the source to sink in this list, adding edges adjacent to the vertex to a min-heap. If the edge is $u \to v$, the value of the edge should be $d(source,u) + w(u \to v) + d(v,sink)$. Here $d(i, j)$ is the shortest distance between $i$ and $j$ while $w(u \to v)$ is the edge weight. Each time $v$ is encountered, the edge is removed from the heap. The minimum distance can be found for each edge along the path by looking at the heap at each vertex level on the shortest path from source to sink.

Problem Code : IOPC1216

## Corruption again[17]

After the first series of protests disrupted the road traffic in Dystopia, the government decided to take counter measures. It chose a few of the roads as high priority roads and sent military vehicles to patrol them. The roads were chosen in a way that it was possible to move from any junction to any other junction just using the patrolled roads. Also, the number of roads chosen was the minimum possible ($N-1$, where $N$ is the number of junctions).

After this, people were able to travel without hindrance from the protestors and the protest wasn't taken seriously anymore. Hence the leader of the protest decided to take things to the next level. He decided that he would personally lead the group which blocks one of the patrolled roads - The military would not dare to touch the group then.

Blocking a patrolled road would divide the road network into two parts $A$ and $B$ such that it would be possible to travel between every pair of junctions in $A$ (and every pair of junctions in $B$) using patrolled roads alone but it would not be possible to move from a junction in $A$ to a junction in $B$ using patrolled roads alone. People then have to take unpatrolled roads with one junction in $A$ and the other in $B$. The leader terms the number of such inter-part roads as the inconvinence factor for the patrolled road being blocked.

Given the road network, find out the maximum inconvenience factor achievable by choosing the patrolled road to block.

### Input

The first line of the input consists of $T$, the number of testcases ($1 \leq T \leq 5$). Following this are the descriptions of the test cases.
The first line of each test case contains two space separated integers $N$ and $M$ - The number of junctions and unpatrolled roads in the network($1 \leq N \leq 10000$; $0 \leq M \leq 30000$). Following this are $N-1$ lines, describing the patrolled roads in the network. Each line will contain two space separated integers $i$ and $j$, denoting that there is a patrolled road between $i$ and $j$. This is followed by $M$ lines describing the unpatrolled roads in a similar fashion. You are assured that there is not more than one direct road between any pair of junctions.

### Output

For each test case, print the maximum value of inconvenience factor achievable among all the possible choices of the patrolled road to be blocked.

[17] Problem formulated by Utkarsh Lath

## Example

### Input:

1
6 3
0 1
1 2
1 3
3 4
3 5
0 2
0 4
4 5

### Output:

2

## Solution

Do a DFS starting from any node. An edge between two vertices $u$ and $v$ will be part of the cut-set only when the edge removed is along the path from $u$ or $v$ to the least common ancestor of $u$ and $v$. So every time a non-tree edge is seen, add 1 to the cut-count for $u$ and $v$, and subtract 2 from the cut-count of the least common ancestor. Then following the reverse dfs order, increment the value of cut-count of every vertex by the cut-counts of its children. The maximum cut-count encountered is the answer we need.