Question 1:

Winter is coming and Jon Snow wants to protect his home from the White Walkers. He has a very small army already divided into battallions. To prepare for any exigency, he wants to know how to quickly divide these battallions further. Since he knows nothing, he needs your help: Given a partitioned set, your job is to print all the immediate refinements of it.

Useful definitions:

A partition of a set is a grouping of the members of the set into non-empty subsets (Recall posets). For example, a possible partition of the set {1, 2, 3} is {{1}, {2, 3}}.

A refinement of a partition of a set is another partition of the same set such that each group of the new partition is a subset of some group of the coarser partition. So a possible refinement of {{1}, {2, 3}} is {{1}, {2}, {3}}.

An immediate refinement of a partition is a refinement which has exactly one more group than the coarser partition. For example, {{1}, {2}, {3}} which has 3 groups is an immediate refinement of {{1}, {2, 3}} which has 2 groups.

Input format:

A number N representing the number of groups in the input partition. Then, for each of these groups, a number K representing the number of elements in the group, followed by K numbers representing the elemnts. For example:

3 // number of groups of the partition
1 // number of elements in the first group
100 //element of the first group
2 //number of elements in the second group
150 200 //elements of the second group
3 //number of elements in the third group
201 202 203 //elements of the third group,

This represents the partition {{100}, {150, 200}, {201, 202, 203}}.

Output format:

Print each possible immediate refinement, one group in one line.

Example:

Input({{1, 2, 3}}):

1
3

1 2 3

Output({{1,3}, {2}}, {{1,2}, {3}}, {{1}, {2, 3}}):

1 3
2
****
1 2
3
****
1
2 3
****

Decoration is optional, but you must make it easy to distinguish between the various immediate refinements.

Question 2:

Our local ninja Naruto is learning to make shadow-clones of himself and is facing a dilemma. He only has a limited amount of energy to spare that he must entirely distribute among all of his clones. Moreover, each clone requires at least a certain amount of energy to function. Your job is to count the number of different ways he can create shadow clones (Check the example for a clearer explanation).

Input format:

Two integers E and M representing the total energy reserve and the minimum energy requirement for a clone. E<50, 0<M<=E

Output format:

One integer representing the number of ways to create clones given the constraints.

Example:

Input:

7 2

Output:

4

The following possibilities occur:

Make 1 clone with 7 energy
Make 2 clones with 2, 5 energy
Make 2 clones with 3, 4 energy
Make 3 clones with 2, 2, 3 energy.

Note: <2, 5> is the same as <5, 2>. Make sure the ways are not counted multiple times because of different ordering.

Input:

10 1

Output:

42

Question 3:

The local mad scientist Qyburn wants to create his own Frankenstein's monster. He already knows what steps he needs to perform, but does not know the order of the steps. And so, he's asking for your help. He knows if a step must necessarily be performed before another, and that's all he tells you. Help him figure out an order for these steps.

Input format:

N, the total number of steps in the first line. Each step is implicitly numbered from 0 to N-1.
A number K in the second line. Then the next k lines contain 2 integers each. Each such line represents a dependancy: If "X Y" is in a line, then step X must necessarily be performed before step Y.

Output format:

One line containing the steps in the correct order. If multiple such correct orders are possible, then print the one in which the smaller numbers come as early as possible.

Example:

Input:

3
2
0 1
0 2

Output:

0 1 2

Explanation:

Step 0 should come before both 1 and 2. So it should be printed first. After step 0, either step 1 or step 2 can be performed, but since we need smaller numbers first, print 1. Then comes 2.

Input:

7
10
2 0
2 1
1 3
3 4
4 0
4 5
0 5
3 6
6 5
1 0

Output:

2 1 3 4 0 6 5

Question 4:

Chi Chi is teaching her son Gohan how to put on a suit. She has given him guidlines of the form "Part A of the suit must be put on before Part B". She claims that her guidlines are thorough and should be enough to exactly teach him how to do it. Gohan is skeptical about this claim and wants to prove her wrong. He wants you to tell him that given the guidlines, how many possible ways of putting on the suit exist.

Input Format:
First line has an integer N, the number of pieces of suit (implicitly numbered from 0 to N-1). The next line contains an integer K. The next K lines have 2 integers each. If a line has X Y, it means that part number X must necessarily be put on before part number Y.

Output Format:
One integer, representing the number of different ways of putting on the suit given the guidlines.

Example:

Input:

5
7
0 1
0 2
0 3
0 4
1 3
1 4
2 4

Output:

5

Explanation:

The following 5 orders are possible:

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

So the answer is 5.

Question 5:

Arya Stark wants to go to King's Landing for an appointment with the queen. The map that she has tells her which cities are reachable from which ones. However, it is not complete. Help her by completing the map. For each city, you must tell her which other cities are reachable based on the information in the incomplete map. (Hint: Think transitivity).

Input Format:

First line contains N, the number of cities (implicitly numbered 0 to N-1). The next line contains K. The next K lines contain 2 distinct integers each. If a line contains X Y, it means Y is reachable from X. Note: A is reachable from be does not imply B is reachable from A.

Output Format:

N lines, each containing a series of integers in ascending order. The series of integers in the kth line represent all the cities reachable from city k. If no city is reachable from a particular city, print "-".

Example:

Input:

7
9
0 1
0 2
2 3
1 4
3 4
3 6
2 6
4 5
5 1

Output:

1 2 3 4 5 6
4 5
1 3 4 5 6
1 4 5 6
1 5
1 4
-

Question 6:

Sherlock Holmes needs your help to solve a case. He has found a lot of evidence, but does not know how to put it together to find a solution. He needs you to write a code to do this for him. He will give you a bunch of axioms of the form (a and b and c and d...) => x where a,b... are atomic prepositions, and some evidence, that is prepositions that are known to be true. Based on these constraints, he wants you to print all the prepositions that are true given the axioms.

Input Format:

First line has N, an integer representing the total number of prepositions (implicitly numbered 0 to N-1). The second line has K, that is, the number of axioms of the form (a and b and c...) => x. The next 2K lines contain duplets of the following form: First line of the duplet contains an integer M, representing the number of conjuncted prepositions in the antecedent. The next line contains M + 1 integers, the first M of them represent the antecedent, the last one the consequent.

After these 2K lines, an integer O, followed by O lines having 1 integer each. The integer represents the prepositions that are set to true. See example

Output:
A list of integers in ascending order representing all the prepositions that are necessarily true given the input.

Example:

Input:

3 // 3 prepositions: 0, 1, 2
2 // 2 implication rules
1 // first rule has 1 antecedent
0 1// first rule says 0 => 1
2 // second rule has 2 antecedents
0 1 2 // second rule says (0 and 1) => 2
,
0 // preposition 0 is true

Output:

0 1 2

Explanation:

0 is given true. Since 0 => 1, 1 is true. Since (0 and 1) => 2, and (0 and 1) is true, 2 is true. Therefore 0, 1, 2 are true.

Input:

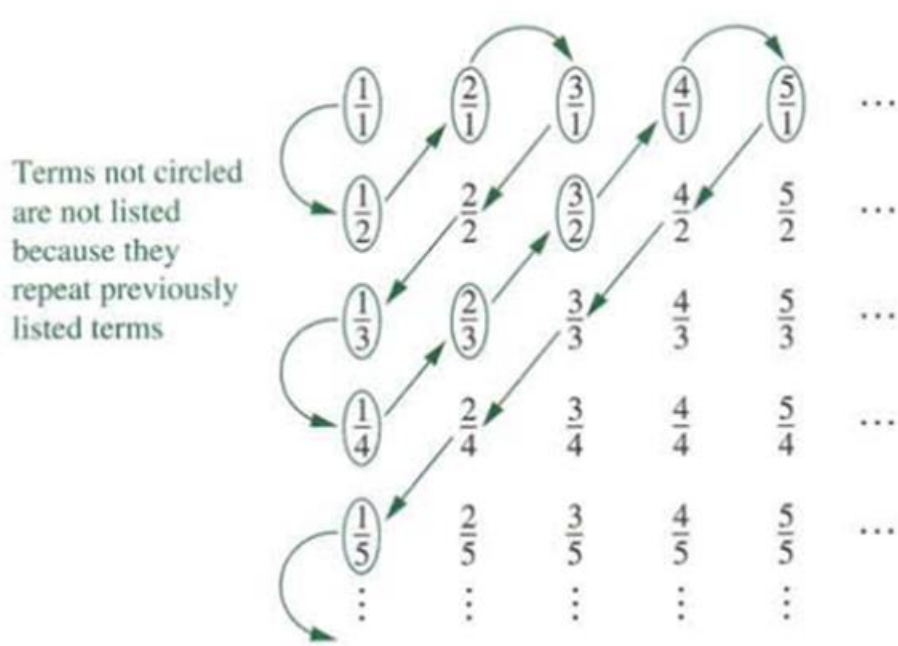
5
4
1
0 3
2
0 2 1
1
1 4
1
4 2
1
0

Output:

0 3

Question 7:

We can call rational numbers as countably infinite because we can define a bijection from the set of rational numbers to the set of natural numbers. We assign rank to the numbers in the set of rational numbers in the following manner. A number is assigned rank only if it's not assigned a rank in it's simplest form. Given a rational number, output it's rank:



Input format:

Two integers p and q to represent the rational number in the form of p/q. You may assume that GCD(p, q) == 1.

Output format:

One integer representing the rank.

Example:

Input:

3 1

Output:

4

Explanation:

The order goes: 1/1 -> 1/2 -> 2/1 -> 3/1. Therefore, 3/1 is ranked 4.

Question 8:

Aang and his friends are sitting in a circle around a fire. His pet Momo, the winged lemur is being passed around by the friends in a clockwise fashion. Each of his friends pushes Momo clockwise by a fixed number of seats. Momo likes only the people who, if they get to pet Momo once, get to pet him again. Help him find the number of people in the circle who satisfy this condition.

Input format:

N, the number of friends in the first line, each implicitly numbered from 0 to N-1 in clockwise fashion. Following N lines have one integer each. The ith of these n lines denotes the number of seats friend i pushes momo clockwise by. That is, if Momo is with friend i, then the next friend who gets Momo is (i + integer in ith of these n lines) % n.

Output format:

One integer, denoting the number of friends in this circle who, if they get Momo once, will get him again eventually.

Example:

Input:

5
1
2
1
2
5

Output:

4

Explanation:

With Friend 0, Momo goes through 0->1->3->0, therefore returns to 0.
Friend 1: 1->3->0->1
Friend 3: 3->0->1->3
Friend 4: 4->4

Hence 4 likeable friends are present.

Question 9:

Uatu and Galactus are hyperdimensional beings. They are bored because the universe is not interesting enough for them, so they decide to play a game. For the game, Uatu summons a magical hypercuboid, with integer lengths of the sides. The rules of the game are as follows: Uatu cuts the hypercuboid into two with a single cut anywhere, making sure that the sides of the resulting two hypercuboids remain integer. Then Galactus picks one of the two pieces, and cuts that into two similarly, leaving three pieces. Next, Uatu picks one of these and performs a similar cut and they keep doing this alternatively. The cuts are all parallel to one of the sides and leave integer-lengthed hypercuboids. The one who cannot make a move loses. As mentioned, Uatu starts the game. Your job is to find who wins, assuming both of them play optimally.

Input format:

N, the number of dimensions in the first line. This is followed by N lines, with an integer each. The integer represents the length of the original hypercuboid in that dimension.

Output format:

"Uatu" or "Galactus" based on who wins. If it is not possible to determine who wins, print "Not sure".

Example:

Input:

2
1
2

Output:

Uatu

Explanation:

Uatu has only one possible choice of cut in a 2x1 rectangle. This leaves two 1x1 squares. Since Galactus can't cut the 1x1 squares any further, Uatu wins.

Input:

5
1
2
3
4
5

Output:

Uatu

Question 10:

Petyr and Varys like to play games with each other. Petyr gives Varys a multiset of numbers. He wants Varys to tell him if it is a "Neat Multiset". A "Neat Multiset" is a multiset in which the products of every pair of elements also exist in the multiset. However, there is a catch. Petyr will only show the numbers to Varys once. That is, Varys cannot write them down. Help Varys identify the "Neat Multiset". **You cannot use an array to store the elements of the multiset.**

Useful definition:

A multiset is a set where the repetition of elements is allowed.

Input format:

N, the number of elements in the multiset. The next N lines contain one integer (can be negative or zero) each, representing the elements of the multiset.

Output format:

Output "Yes" if the multiset is "Neat". Output "No" if it is not. Output "Cannot tell" if it is impossible to find out given the no-array constraint.

Example:

3
0
1
2

Output:

Yes

Explanation:

0x1 = 0
0x2 = 0
1x2 = 2

0 and 2 are present in the multiset. So this multiset is "Neat".

Input:

3
1
2
3

Output:

No

Question 11:

Saitama is very picky about his sets. He only likes his sets to be "Serious" (See below for definition). He sends his sidekick Genos to get him a non-empty "Serious Set". However, Genos only has a certain set of integers to make a Serious Set out of. Help Genos find all the non-empty "Serious Sets" that he can make. Print all the "Serious Sets" that are a non-empty subset of the set that Genos has already.

Definition:

A "Serious Set" is a set in which the number of possible pairs (a, b) from the set where b divides a is even if the cardinality of the set is even, and odd if the cardinality of the set is odd. Note that a is not equal to b. Formally, a set S is serious if: is_even(| {(a, b)| a, b belong to S AND b divides a AND a is not equal to b} |) <=> is_even(| S |) AND | S | is not 0.

Input format:

First line has N, the number of elements in the set that Genos has. The next N lines have 1 integer each, representing the elements of the set that Genos has.

Output format:

Each possible "Serious Set", as space separated integers, on separate lines.

Example:

Input:

4
1
2
3
4

Output:

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

Explanation:

{2, 3} has 0(even) pairs and cardinality of {2, 3} is also even.
{1, 2, 4} has 3(odd) pairs (2, 1), (4, 1), (4, 2) and cardinality of {1, 2, 4} is also odd.
{3, 4} has 0(even) pairs and cardianlity of {3, 4} is also even.
{2, 3, 4} has 1(odd) pair (4, 2) and cardinality of {2, 3, 4} is also odd.
{1, 2, 3, 4} has 4(even) pairs (2, 1), (3, 1), (4, 1), (4, 2) and cardinality of {1, 2, 3, 4} is also even.

Input:

5
1
2
3
4
5

Output:

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

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

Question 12:

The Avengers have acquired the Tesseract from Loki. It turns out that the Tesseract is actually a divisor lattice (remember from class that a divisor lattice of a number is a lattice of all the factors of the number with the partial ordering "a divides b"). To measure the power of the Tesseract, they need to know how many edges it has (An edge in a lattice is a connection between two elements) for which they require your help. Given a number, find out the number of edges in its divisor lattice.

Input format:

One integer representing the number.

Output format:

One integer representing the number of edges in the divisor lattice of the input.

Example:

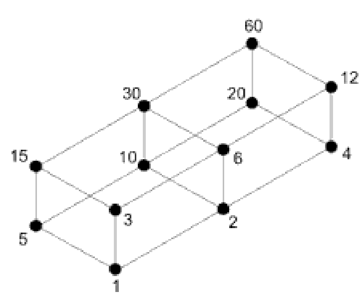
Input:

60

Output:

20

Explanation:



The number of edges is 20.

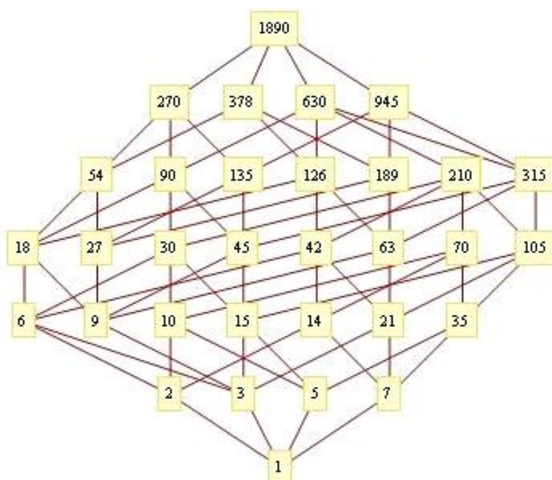
Input:

1890

Output:

72

Explanation:



The number of edges is 72.

Question 13:

Tyrion Lannister loves relations. In particular, he loves reflexive relations. His friend Bronn gives him an incomplete relation, and tells him that it is transitive in nature. Tyrion wants to know if it is reflexive or not, so he asks for your help. The relation is defined from a set to itself.

Input format:

N, the number of elements in the set, implicitly labeled 0 to N - 1. Next line has K. The next K lines have two integers each. If a line has "x y", it means that (x, y) is in the relation.

Output format:

"Reflexive" if the relation is reflexive, "Not reflexive" otherwise.

Example:

Input:

3
4
0 0
1 0
0 1
2 2

Output:

Reflexive

Explanation:

(0, 0) and (2, 2) are in the relation. (1, 0) and (0, 1) are in the relation. Since the relation is transitive, (1, 1) must also be in the relation. Therefore, since (0, 0), (1, 1) and (2, 2) are all in the relation, the relation is reflexive.

Question 14:

The Avengers have found another Tesseract. This tesseract is actually a subset lattice (recall that a subset lattice of a set is a lattice of all possible subsets of the set, ordered by the partial order "is a subset of"). Yet again they need to calculate the number of edges the lattice has to gauge its power. Help them do that.

Input format:

One integer, representing the number of elements of the set.

Output format:

One integer, representing the number of edges in the subset lattice of the set.

Example:

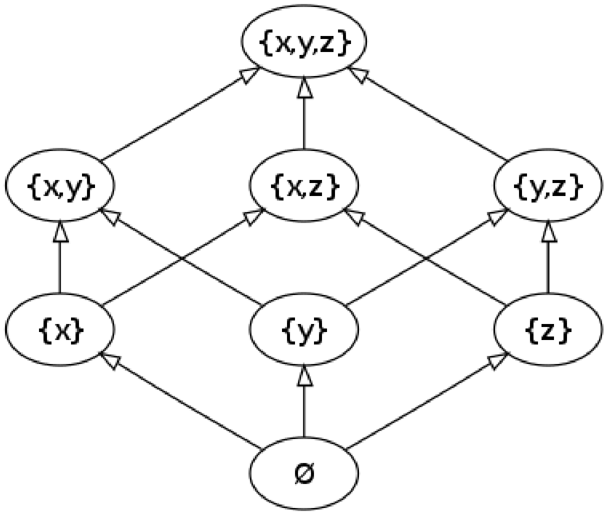
Input:

3

Output:

12

Explanation:



A three element set has a subset lattice with 12 edges.

Question 15:

Ned Stark is a noble king and wants order (at least partial) in the relations of his kingdom. His friend Robert gives him a new relation. Help Ned decide if he should allow the relation to stay in his kingdom or not by telling him if it is a partial order relation or not.

Input format:

An integer N in the first line, representing the number of elements in the set on which the relation is defined. The elements are implicitly numbered from 0 to N-1. Next line has an integer K. Next K lines have two integers each. If "x y" are the two integers, it means that the relation has the duplet (x, y).

Output format:

"Yes" if the given relation is a partial order relation, "No" otherwise.

Example:

Input:

3
6
0 0
0 1
0 2
1 1
1 2

2 2

Output:

Yes

Question 16:

Saitama is the strongest hero ever, defeating all his enemies with a single punch. His disciple Genos wants to become strong like him. To this end, Saitama teaches Genos a few exercises. However, he warns that certain exercises *must* be performed before others. Since Saitama is a bit crazy, Genos wants to make sure that performing the exercises is actually possible given these constraints. Help Genos figure out if an actual order of the exercises exists.

Input format:

First line has an integer N, the number of exercises implicitly labeled 0 to N-1. The second line has an integer K. The next K lines have 2 distinct integers each. If a line has "x y", then exercise x must be performed before exercise y.

Output format:

"Possible" if an order fitting the constraints exists, "Not Possible" otherwise.

Example:

Input:

2
2
0 1
1 0

Output:

Not Possible

Explanation:

You have to do exercise 0 before exercise 1 and you have to do exercise 1 before exercise 0. Therefore, no order fitting the constraints exists.

Input:

3

3
0 1
0 2
1 2

Output:

Possible

Explanation:

A possible order is 0->1->2.

Question 17:

The X-Men are training and this time, Professor X wants to train them in pairs. Each of them has a set of skills out of a pool of all possible skills a mutant can have. Professor wants to find the best partner for the new student Colossus while at the same time, wants to achieve a particular skill level of the combined pair. Help Professor X list out all the possible skillsets that in the partner that can achieve this. Given a set S of all skills possible, one of its subsets T (the required skills in the pair) and one of this subset's subset U (Colossus's skills), list all the possible subsets of S such that the union of any one of these subsets with U gives T.

Input format:

First line has N, the number of all possible skills. The next N lines have 1 integer each, representing the skill label. The next line has K, the number of skills that Colossus has. The next K lines have 1 integer each, representing the labels of Colossus's skills. The next line has L, the number of skills in the required set. The next L lines have 1 integer each, representing the labels of these required skills.

Output format:

Each possible set, represented by space space separated integers (in any order), union of this set with the skill set of Colossus is the required set.

Example:

Input:

5
1 2 3 4 5
2
1 2
4

1 2 3 4

Output:

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

Explanation:

S = {1, 2, 3, 4, 5}, U = {1, 2}, T = {1, 2, 3, 4}. Union of any of {3, 4}, {1, 3, 4}, {2, 3, 4} or {1, 2, 3, 4} with {1, 2} yields {1, 2, 3, 4}.

Question 18:

Batman has a large collection of gadgets in his Batcave. However, owing to budget constraints, his new belt can hold only a certain number of them. Batman is getting ready to fight the Joker in the Crime Alley and is confused about what gadgets he should take with him. He wants you to list all possible combinations out for him so that he can decide.

Input format:

N, the number of gadgets. The next N lines contain one integer each, representing the gadget labels. The next line contains a number K, the capacity of the belt.

Output format:

All possible combinations of K gadgets from N, as space separated integers (order does not matter). Each combination to be printed in a separate line.

Example:

Input:

5
1
2
3
4
5
3

Output:

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

Question 19:

Given a number, check if the number is a "Good number". A "Good number" is defined as a number which, in its binary representation, has the same number of ones as it's complement.

Input format:

N, the number to be tested.

Output format:

"Yes" if it is a good number, "No" otherwise.

Example:

Input:

2

Output:

Yes

Input:

1

Output:

No

Question 20:

Given 2 numbers M and N, find how the unit's place digit of the expression M^N.

Input format:

M N, two numbers in a line.

Output format:

One integer representing the unit's place digit of M^N.

Example:

Input:

104 3

Output:

4

Explanation:

104^3 = 1124864. Therefore, the answer is 4.

Input:

2023 1917

Output:

3