GRAN PREMIO 2017

PRIMERA FECHA

Problem A. Tobby's Ancestors

Source file name:	ancestors.c, ancestors.cpp, ancestors.java, ancestors.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Sebastián Gómez - UTP Colombia`

Tobby, the small and cute dog, wants to prove to the rest of the world he is a descendent of Tutantobby, the great dog pharaoh. As Tutantobby was mummified, his internal organs were dried and separated in different bottles. To prove that Tobby is a descendent of Tutantobby, the DNA of some of these organs must be extracted and compared against Tobby's DNA. This DNA comparison is not a problem for nowadays science, but extracting the DNA from a 5000 years old mummified organ is a real challenge.

The DNA, as you might probably know, is represented as a sequence of letters `A`, `G`, `C` and `T`. All Tobby can expect is to obtain fragments of Tutantobby's DNA corrupted by the years. Tobby requires to assemble Tuntantobby DNA fragments, to do that Tobby takes two DNA fragments and computes the best way to assemble them as in the following example. Lets say Tobby has the two following Tutantobby DNA segments:

- `GATTACCA`
- `TACAACAG`

Note the following alignment and the resulting string:

```
GATTACCA
   TACAACAG
------------
GATTACXACAG
```

The score of this alignment is 4, since there are 4 characters in this alignment that match. The resulting assembled DNA would then be `GATTACXACAG`, note that the resulting string might have an X letter representing that Tobby can't tell what that letter would be. Now Tobby wants a program that given only two sequences, outputs the assembled DNA sequence for the alignment with the maximum score. If there are two alignments that produce the same score output the one where the first string is more to the left with respect to the second. For example if the first string is `ATTG` and the second is `GCCA`, the alignment `ATTGCCA` is preferable over `GCCATTG` since both have a score of 1.

Input

The input consists of several test cases. Each test case contains two strings S_1 and S_2 that indicate the two DNA fragments extracted from Tutantobby. These two strings will only contain the uppercase letters `A`, `G`, `T` and `C`. After each test case there will be a blank line.

- $1 \leq |S_1|,\ |S_2| \leq 10^5$

Output

For each test case output two lines. The first one with the best alignment score, and on the second line output the respective assembled sequence as explained in the problem statement. If the best score is 0, print on the second line the string `No matches`. There should be a blank line after each test case.

Example

Input	Output
GATTACCA TACAACAG AAAA GGGG ATTG GCCA	4 GATTACXACAG 0 No matches 1 ATTGCCA

Use fast I/O methods

Problem B. Tobby and Array

Source file name:	tobbyarray.c, tobbyarray.cpp, tobbyarray.java, tobbyarray.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Jhon Jimenez, Manuel Pineda & Santiago Gutierrez - UTP Colombia`

As it is known, Tobby loves arrays and queries (he also hates long statements :D). One day Tobby came up with the following: there is an array of integers and multiple queries. For each query, Tobby wants to know the value of the $k-th$ position in the subarray $[l,\ r]$ $(r \geq l)$ $(1 \leq k \leq r-l+1)$, **if the subarray [l, r] was sorted in non-decreasing order**.

Input

The input has several test cases. The first line contains n $(1 \leq n \leq 10^6)$ and $q(1 \leq q \leq 10^6)$, the length of the array and the number of queries respectively. The next line contains n integers $a_i(1 \leq a_i \leq 10^9)$. Then q lines follow, each line containing a query with three integers l, r and k $(1 \leq l,\ r \leq n)$.

Output

For each query print the answer in a single line (**Look at the samples**).

Example

Input	Output
`4 3`	`3`
`1 3 4 3`	`3`
`1 2 2`	`4`
`2 4 1`	`3`
`1 4 4`	`3`
`8 3`	`8`
`4 7 8 5 3 6 1 2`	`3`
`4 5 1`	`10`
`1 8 3`	`9`
`3 5 3`	`5`
`10 10`	`10`
`8 6 2 1 7 3 10 9 5 4`	`2`
`1 8 3`	`3`
`7 7 1`	`4`
`7 8 1`	`5`
`9 9 1`	`10`
`2 10 9`	
`2 7 2`	
`5 7 1`	
`10 10 1`	
`9 10 2`	
`7 10 4`	

Use fast I/O methods

Explanation

For the first sample.

$indexes$: 1 2 3 4
$array = \{1,\ 3,\ 4,\ 3\}$

For first query $[1,\ 2]$ we have the subarray $\{1,\ 3\}$, after sorting we have $\{1,\ \overline{3}\}$, so the value in the $2-th$ position is 3.

For second query $[2,\ 4]$ we have the subarray $\{3,\ 4,\ 3\}$, after sorting we have $\{\overline{3},\ 3,\ 4\}$, so the value in the $1-th$ position is 3.

For third query $[1,\ 4]$ we have the subarray $\{1,\ 3,\ 4,\ 3\}$, after sorting we have $\{1,\ 3,\ 3,\ \overline{4}\}$, so the value in the $4-th$ position is 4.

Problem C. Counting Edges and Graphs

Source file name:	counting.c, counting.cpp, counting.java, counting.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Yonny Mondelo - UCI Cuba`

You must construct a directed graph with exactly N nodes conveniently numbered between 1 and N. But this is not an ordinary graph; this is a special graph for which each node must have K or less directed edges going to their proper divisors (nodes numbered with those divisors values). If some node has only $k \leq K$ proper divisors, then that node will have exactly k edges to those k divisors. Also note that if some node has $M > K$ proper divisors then that node will have exactly K edges to some group of K proper divisors of the M available. A proper divisor of some integer number P is any divisor of P, excluding P itself. For example, 1, 2 and 3 are proper divisors of 6; but 6 is not a proper divisor of itself.

Given the value for K and the number of nodes N in the graph you must construct, can you find the number of edges on it after it is constructed? Also, can you determine the number of possible graphs which can be constructed fulfilling the above specifications?

Input

The first line contains an integer T ($4*10^5 \leq T \leq 5*10^5$) representing the number of graphs to construct. The next T lines contain two integer numbers N and K ($1 \leq N,\ K \leq 5*10^3$) representing the number of nodes in the graph and the maximum number of edges per node. Scenarios must be answered in the same order of the graphs given in the input.

Output

For each graph you must print a line containing two integer numbers representing the number of edges of the graph and the number of possible graphs which can be constructed, respectively. As those values could be large, print them modulo 1000000007 ($10^9 + 7$).

Example

Input	Output
3 4 2 5 3 6 2	4 1 5 1 7 3

Use fast I/O methods

Problem D. DPA Numbers I

Source file name:	dpa01.c, dpa01.cpp, dpa01.java, dpa01.py
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

In number theory, a positive integer belongs to one and only one of the following categories: Deficient, Perfect or Abundant (DPA).

To decide the category of a positive integer n, first you have to calculate the sum of all its proper positive divisors. If the result is less than n then n is a deficient number, if the result is equal to n then n is a perfect number and if the result is greater than n then n is an abundant number. Remember that the proper divisors of n don't include n itself.

For example, the proper divisors of the number 8 are 1, 2 and 4 which sum 7. Since $7 < 8$ therefore 8 is a deficient number. The proper divisors of the number 6 are 1, 2 and 3 which sum 6. Since $6 = 6$ therefore 6 is a perfect number. The proper divisors of the number 18 are 1, 2, 3, 6 and 9 which sum 21. Since $21 > 18$ therefore 18 is an abundant number.

The task is to choose the category of a positive integer n as a deficient, perfect or abundant number.

Input

Input begins with an integer t ($400 \leq t \leq 500$), the number of test cases, followed by t lines, each line containing an integer n ($2 \leq n \leq 10^3$).

Output

For each test case, you should print a single line containing the word `deficient`, `perfect` or `abundant` that representing the category of the number n.

Example

Input	Output
10 5 6 16 18 21 28 29 30 40 43	deficient perfect deficient abundant deficient perfect deficient abundant abundant deficient

Problem E. Tobby and the quaseEquals strings

Source file name:	tobbystrings.c, tobbystrings.cpp, tobbystrings.java, tobbystrings.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Carlos Arias - UTP Colombia`

Tobby always enjoys playing with strings, and now he brings to you a nice problem with them. Of course, since Tobby is a lazy dog, he has not solved it yet and hopes that you can solve it for him.

Tobby got a set of strings S of size N (where every string has the same length L). He also has Q queries.

For each query a string A of size L is given and Tobby wants to know how many strings in S are quaseEquals to A for every i $(1 \leq i \leq L)$.

Two strings are quaseEquals to one another for an index i if they are equal after deleting the i-th character from both strings.

Input

The input consists of several test cases, read until the end of file (EOF). In the first line of each test case there are three integers: N, Q, L $(1 \leq N,\ Q,\ L \leq 10^5)$. The next N lines contain the strings in S, all of length L. Finally Q strings of length L are given, those are the queries. It is guaranteed that $(1 \leq N * L \leq 100000)$ and $(1 \leq Q * L \leq 100000)$ and that all strings in the input contain only english lowercase letters (a-z).

Output

For each query print the number of strings in S that are quaseEquals to the string in the query for every position $1 \leq i \leq L$.

Example

Input	Output

`3 1 3`
`aab`
`aba`
`aaa`
`aaa`
`6 3 6`
`tobyis`
`having`
`funwhi`
`leyoua`
`resolv`
`ingitD`
`tobbis`
`cobyis`
`cobbis`
	`5`
	`1`
	`1`
	`0`

Use fast I/O methods

Explanation

For the first sample, if the character $i = 1$ is removed, then $S = \{ab,\ ba,\ aa\}$ and $A = \{aa\}$ and we got 1 pair of quaseEquals strings. If the character $i = 2$ is removed, then $S = \{ab,\ aa,\ aa\}$ and $A = \{aa\}$ and we got 2 pairs of quaseEquals strings. If the character $i = 3$ is removed, then $S = \{aa,\ ab,\ aa\}$ and $A = \{aa\}$ and we got 2 pairs of quaseEquals strings, so our answer is $1 + 2 + 2 = 5$.

Problem F. Felipe and the Sequence

Source file name:	sequence.c, sequence.cpp, sequence.java, sequence.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

On February 19, 2017, Red Matemática proposed the following mathematical challenge on its twitter account (@redmatematicant): "Felipe, how many terms of the next sequence of numbers must be added to make the result equal to 200?"

$$\frac{1}{\sqrt{1}+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + \frac{1}{\sqrt{3}+\sqrt{4}} + \frac{1}{\sqrt{4}+\sqrt{5}} + \cdots = 200$$

Using this interesting puzzle as our starting point, the problem you are asked to solve now is: Given a positive integer S $(1 \leq S \leq 10^9)$ representing the result obtained for the sum of the terms in the sequence, find out the number n that represents the total number of terms in the sequence to sum up.

$$\frac{1}{\sqrt{1}+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + \frac{1}{\sqrt{3}+\sqrt{4}} + \frac{1}{\sqrt{4}+\sqrt{5}} + \cdots + \frac{1}{\sqrt{n}+\sqrt{n+1}} = S$$

Input

Input begins with an integer t $(4 * 10^5 \leq t \leq 5 * 10^5)$, the number of test cases, followed by t lines, each containing an integer S $(1 \leq S \leq 10^9)$.

Output

For each test case, your program must print one positive integer denoting the number n that represents the total number of terms in the sequence to sum up.

Example

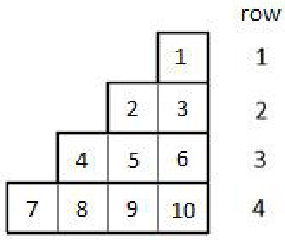
Input	Output
3 200 725 1333	40400 527075 1779555

Use fast I/O methods

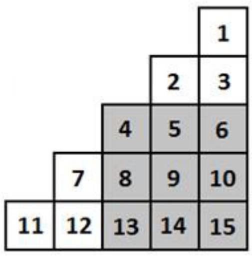
Problem G. Rectangular Sum

Source file name:	rectangular.c, rectangular.cpp, rectangular.java, rectangular.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gabriel Gutiérrez Tamayo - UTP Colombia`

In this challenge, you are given a triangular board of n rows. The first row has one block, and the following rows have a block more than the previous row. All the blocks have the same size and are numbered as follows:



First, you must find the biggest rectangular area inside the triangular board, and then calculate the value of S which corresponds to the sum of the values belonging to the area found. If there are several areas with the same size, choose the area that maximizes the value of S. For example, when $n = 5$:



The maximum rectangular area is (3×3), which is represented in the previous image.

$S = 4 + 5 + 6 + 8 + 9 + 10 + 13 + 14 + 15 = 84$

Remember that the area of a rectangle is the multiplication of the two sides of the rectangle

Input

The first line of input contains an integer t ($10^4 \le t \le 10^5$) indicating the number of test cases that follow, one for line. Each test case contains a positive integer n ($1 \le n \le 10^{11}$) indicating the number of rows.

Output

For each test case, you should print a line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the sum obtained. Note that this value is very large, so print the result modulo $10^9 + 7$.

Example

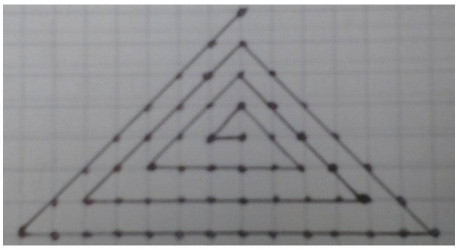
Input	Output
6 1 2 3 4 5 14	Case #1: 1 Case #2: 5 Case #3: 16 Case #4: 42 Case #5: 84 Case #6: 3612

Use fast I/O methods

Problem H. Humbertov and the Triangular Spiral

Source file name:	triangular.c, triangular.cpp, triangular.java, triangular.py
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

Recently the professor *Humbertov Moralov* was sick, he had a fever and when he went to bed, he began to have a delirious dream. In the dream he draw over and over again a triangular spiral that began in the origin of the cartesian plane (coordinate $(0, 0)$) and the triangular spiral got bigger every time linking integer coordinates in the cartesian plane. For clarity, the triangular spiral is presented below:



The dream was so disturbing and it was repeated so many times, that when *Moralov* woke up, he remembered perfectly the triangular spiral, and for this reason he drew the previous graphic.

In the dream *Moralov* was disturbed and intrigued because he didn't know if all the integer coordinates could be reached at some point in the triangular spiral and, if that was the case, he also didn't know what would be the coordinate in the cartesian plane of the n-th point that is reached when drawing the triangular spiral. The first doubt was immediately resolved when the professor did the graphic ... all the points (integer coordinates) of the cartesian plane are eventually reached by the triangular spirals! Now the professor *Moralov* needs your help to indicate the coordinate in the cartesian plane of the n-th point that is reached when drawing the triangular spiral.

Input

Input begins with an integer t ($4 \times 10^5 \leq t \leq 5 \times 10^5$), the number of test cases, followed by t lines, each line contains an integer n ($1 \leq n \leq 10^{12}$).

Output

For each test case, you should print a single line containing two integers, separated by a space, denoting the coordinates x and y in the Cartesian coordinate system of point n in the triangular spiral.

Example

Input	Output
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0 0 -1 0 0 1 1 0 2 -1 1 -1 0 -1 -1 -1 -2 -1 -3 -1 -2 0 -1 1 0 2 1 1 2 0

Use fast I/O methods

Problem I. Rockabye Tobby

Source file name: rockabye.c, rockabye.cpp, rockabye.java, rockabye.py
Input: `Standard`
Output: `Standard`
Author(s): `Yeferson Gaitan Gomez - UTP Colombia`

"Rockabye baby, don't you cry".

Tobby is very good at catching the ball, he loves that game so much, that one day he decided to go out and play, even though it was raining. He played for a long time and in addition to catching the ball many times, he also got a cold, poor Tobby. That is why now his mother will take care of him, Big doggie momma, singing that beautiful lullaby (rockabye) and giving him the medications in the moments that must be taken.

In the medical prescription sent by the doctor, he specifies the name of the medications and how often they should be taken. The doctor told him that if he takes the medications constantly, he will be relieved after taking k medicines. Tobby does not like being sick (in fact no one likes to be), so he promises his mother to be constant with the drugs, that is why he now wants to know what are the first k drugs that he has to take to feel better. Can you help him?

Input

Input begins with a line containing an integer t, the number of test cases.

For each test case, the medical prescription is written as follows:

Each test case begins with a line containing two integers, n $(1 \leq n \leq 3 * 10^3)$ and k $(1 \leq k \leq 10^4)$, indicating the number of medications sent by the doctor and the minimum number of medicines Tobby must take to feel better.

The following n lines will be of the form, `name frecuency` $(1 \leq |$ `name` $| \leq 10, \quad 1 \leq$ `frecuency` $\leq 3 * 10^3)$, indicating the name of the medication and how often it should be taken.

The medicines are listed according to their degree of priority, i.e. the first one will be the most important drug and the last one, the least important.

Output

For each test case, the output must have k lines, each of the form, t m, indicating that in the moment t Tobby must take the drug m.

If there are two or more drugs that must be given at the same time t, they should be printed according to their priority.

Example

Input	Output
1 2 5 Acetaminophen 20 Loratadine 30	20 Acetaminophen 30 Loratadine 40 Acetaminophen 60 Acetaminophen 60 Loratadine

Use fast I/O methods

Problem J. Tobby Primes

Source file name:	tobbyprimes.c, tobbyprimes.cpp, tobbyprimes.java, tobbyprimes.py
Input:	Standard
Output:	Standard
Author(s):	Santiago Gutierrez - UTP Colombia & Google

Tobby the boston terrier is trying to escape from the pyramid of the egyptian pharaoh. To escape, Tobby has to solve a riddle that asks him to factor a list of large integers.

The fastest way Tobby knows of factoring integers is iterating over all primes up to the square root of the target number, checking for prime factors. The problem is that in this case the number of primes to check would be very large, so he cannot use that method. Can you help him solve the riddle?

Input

Input begins with an integer t ($1 \leq t \leq 100$), the number of integers to factor, followed by t lines, each line contains an integer n ($2 \leq n \leq 2^{63} - 1$).

Output

For each test case, you should print a single line containing the prime factors of n sorted in increasing order. Note that in case of repeated prime factors, such factors have to be printed several times.

Example

Input	Output
3 2 91 40	2 7 13 2 2 2 5

Problem K. Tobby and the Skeletons

Source file name:	tobbyskeletons.c, tobbyskeletons.cpp, tobbyskeletons.java, tobbyskeletons.py
Input:	`Standard`
Output:	`Standard`
Author(s):	`Diego Agudelo-España - UTP Colombia`

There is nothing that Tobby, as a dog, enjoys more than bones. That's why he has been studying the skeleton of certain species to figure out how much fun he could have with their bones.

Tobby models a skeleton as a weighted tree whose edges represent the bones and their weights denote the lengths of the bones. Thus, the nodes are simply the joints connecting different bones. For Tobby it is quite hard to play with an entire skeleton, so he prefers to take a chain of connected bones instead, or, in other words, a simple path connecting two nodes in the skeleton tree. Moreover, since Tobby is a greedy dog, his happiness with a particular chain of bones doesn't depend only on the chain's size but on the length of the largest bone present in the chain as well, and here is where Tobby needs some help from you.

It turns out that even for members of the same species there are variations on the length of the bones that make up the skeleton (the skeleton itself keeps fixed across members). Therefore, Tobby decided to model the bone lengths (i.e. edge weights) as discrete uniform random variables. This means that the weight w_i associated with the $i-th$ edge takes integer values in the closed interval $[a_i,\ b_i]$.

Tobby has Q queries for you. Given a description of the tree and the weight random variables, for each query Tobby wants to know the expected value over the length of the largest bone present in a bones chain from joint x_q to join y_q. Formally, if $w_1,\ w_2,\ \ldots,\ w_s$ are the random variables denoting the edge's weights in the simple path from x_q to y_q, you are required to compute $E[max(w_1,\ w_2,\ \ldots,\ w_s)]$.

Input

The input contains multiple test cases. For each test case the first line contains an integer N $(2 \leq N \leq 50000)$ denoting the number of nodes in the tree of bones. Each of the following $N-1$ lines describes an edge with 4 integers: x_i, y_i, a_i, b_i $(x_i \neq y_i,\quad 1 \leq x_i,\ y_i \leq N,\quad 0 \leq a_i \leq b_i \leq 100)$ where x_i and y_i represent two different nodes connected by the edge whose weight can take discrete values uniformly in $[a_i,\ b_i]$ (It's guaranteed that the given graph is a tree). Next, the number of queries Q is given and then, Q $(1 \leq Q \leq 100000)$ more lines follow describing each query with two different nodes x_q and y_q $(x_q \neq y_q,\quad 1 \leq x_q,\ y_q \leq N)$ which represent both ends in a bones chain of interest for Tobby. The input specification ends with EOF.

Output

For each test case there must be Q output lines answering the Q test case queries. For each of these queries print in a single line the expected value of the largest edge weight in the simple path connecting the queried nodes. Your answer will be considered correct if the absolute difference with the jury's answer is less than $1e^{-5}$.

Example

Input	Output
6 1 2 0 50 2 3 30 40 2 4 10 80 4 5 50 90 4 6 95 100 3 1 6 3 5 2 4 2 1 2 0 100 1 2 1	97.5 71.70388182755067 45.000000000000014 49.999999999999986

Use fast I/O methods

Problem L. Rotations

Source file name:	rotations.c, rotations.cpp, rotations.java, rotations.py
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

Humbertov Moralov in his student days, enrolled in the Systems Engineering program at "University of the Missing Hill" in *The Heaven's Branch Office* (Colombia, South America). He then attended a course of Assembly Language (in the first half of 1997).

The course was fantastic, with very interesting topics such as bit manipulation, right shifts, left shifts, rotations, masks and other bitwise operations (and, or, xor, not). And the best, in that course he worked interesting programming challenges. One of those programming challenges follows.

The chosen programming challenge is named "Rotations". By that time he used to work with unsigned integers of eight bits (a byte), and the challenge consisted of figuring out if a particular number n could generate all the eight numbers from 0 to 7 taking groups of consecutive bits of size 3.

For example, the number 226 has the binary representation 11100010 ($b_7b_6b_5b_4b_3b_2b_1b_0$), the eight sequences of consecutive three-bits that can be generated are the following:

- $b_2b_1b_0 = (010)_2 = 2$
- $b_3b_2b_1 = (001)_2 = 1$
- $b_4b_3b_2 = (000)_2 = 0$
- $b_5b_4b_3 = (100)_2 = 4$
- $b_6b_5b_4 = (110)_2 = 6$
- $b_7b_6b_5 = (111)_2 = 7$
- $b_0b_7b_6 = (011)_2 = 3$
- $b_1b_0b_7 = (101)_2 = 5$

20 years have passed. Since today computers are more powerful and faster, the professor *Humbertov Moralov* wants you to solve this programming challenge for unsigned integers of 32 bits (four bytes). You must validate if the number n generates all the 32 numbers from 0 to 31 with sequences of consecutive five-bits.

Input

Input begins with an integer t ($2 \times 10^5 \leq t \leq 3 \times 10^5$), the number of test cases, followed by t lines, each line containing an integer n ($0 \leq n \leq 4 \times 10^9$).

Output

For each test case, you should print a single line containing the word `yes` or `no` depending if the integer number n produces or not all the numbers from 0 to 31 with sequences of consecutive five-bits.

Example

Input	Output
15	no
65535	no
65259922	yes
81354525	no
112805325	no
122525196	yes
192052550	yes
225525450	no
299525510	yes
318353525	no
344152934	yes
502445252	yes
522595252	no
1296752550	yes
3999995011	no
4000000000	

Use fast I/O methods

Problem M. DPA Numbers II

Source file name:	dpa02.c, dpa02.cpp, dpa02.java, dpa02.py
Input:	`standard`
Output:	`standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

In number theory, a positive integer belongs to one and only one of the following categories: Deficient, Perfect or Abundant (DPA).

To decide the category of a positive integer n, first you have to calculate the sum of all its proper positive divisors. If the result is less than n then n is a deficient number, if the result is equal to n then n is a perfect number and if the result is greater than n then n is an abundant number. Remember that the proper divisors of n don't include n itself.

For example, the proper divisors of the number 8 are 1, 2 and 4 which sum 7. Since $7 < 8$ therefore 8 is a deficient number. The proper divisors of the number 6 are 1, 2 and 3 which sum 6. Since $6 = 6$ therefore 6 is a perfect number. The proper divisors of the number 18 are 1, 2, 3, 6 and 9 which sum 21. Since $21 > 18$ therefore 18 is an abundant number.

The task is to choose the category of a positive integer n as a deficient, perfect or abundant number.

Input

Input begins with an integer t ($1000 \leq t \leq 1100$), the number of test cases, followed by t lines, each line containing an integer n ($2 \leq n \leq 10^{12}$).

Output

For each test case, you should print a single line containing the word `deficient`, `perfect` or `abundant` that representing the category of the number n.

Example

Input	Output
10	deficient
5	perfect
6	deficient
16	abundant
18	deficient
21	perfect
28	deficient
29	abundant
30	abundant
40	deficient
43	

Use fast I/O methods

GRAN PREMIO 2017 SEGUNDA FECHA

Problem A. Adventure on the space

Source file name:	`A.c, A.cpp, A.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI Guadalajara`

Professor JJ has developed the most powerful space travel machine, this machine has the ability it can go from planet A to planet B in just a few miliseconds if B is in the spatial range from A. The spatial range in a planet is given by an integer D, and it is said B is in the spatial range of A if the distance between the planets is at most D.

As some of you are aware JJ prefers always to go in the most secure way, this is why he is developing what he calls the most secure grid of planets. The most secure grid of planets is created selecting pairs of planets where you could travel between those planets using JJ's space travel machine and also it follows these restrictions:

- All planets in the universe should be in the grid of planets.
- For any pair of planets A and B there is only one way to reach B from A using JJ's space travel machine

You always argue with JJ, and now you two are arguing that he should not call his grid "the most secure grid of planets" as there can be more than one way to create it or there may not be a way to create the grid at all. JJ will not believe you until you get some evidence. This is why he just challenged you (as always) given the number N of planets in the universe, the value D that defines the spatial range of a planet and the coordinates where the planets are in the universe, can you determine how many different "secure grids" JJ can create? A "secure grid" A differs from other "secure grid" B if there is at least one pair of planets in the grid A that is not in the grid B.

Input

The input consist of several test cases. Each test case begins with a line containing two numbers N and D. The next N lines contains the values x_i, y_i separated by a space the coordinates of each planet. The end of the test cases is given by a case where N and D equals 0.

- $2 \leq N \leq 10$
- $1 \leq D \leq 10^3$
- $0 \leq x_i, y_i \leq 10^3$

Output

For each test case print in one line the number of different "secure grids" JJ can create. As this number can be very large please print it modulo $10^9 + 7$

Example

Input	Output
3 5 1 1 2 2 1 2 3 1 1 1 2 2 1 2 0 0	3 0

Explanation

In the first case, there are 3 planets and you can reach any of them from any other. Let the planets be 1,2,3 , you can select the pairs to create 3 different “secure grids” : 1,2,2,3, 1,2,1,3, 1,3,2,3. In the second test case there is no way to select pairs in such a way that all the requirements for a “secure grid” are met.

Problem B. Balloons

Source file name:	`B.c, B.cpp, B.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Félix Arreola - CUCEI Guadalajara`

This year CUCEI's programming contest is so easy that Félix believes every team will solve all the problems, these are good news for you but not for Félix. As you may know each problem has assigned a color, once a team solves that problem a balloon with that color is put in the place where the team who solved the problem is working, since all the teams will be solving all the problems Félix needs to get each balloon color as much times as teams registered in the contest.

The budget for the contest is very low and Félix will feel so bad if a team can't have all the balloons of the problems the team solved. This left Félix with no choice but to limit the number of teams that will be allowed in the contest with a simple rule. If Félix can buy only X balloons of each color with the budget Félix has then only X teams will be allowed to register on the contest.

Given the budget B Felix has, the number N of problems in the set and the cost of each color of balloon, can you determine what is the maximum number of teams that Félix can allow in the contest?

Input

The input consist of several test cases. Each test case begins with a line containing two numbers N and B. The next line contains N numbers separated by a space, the cost of each of the balloon colors. The end of the test cases is given by a case where N and B equals 0.

- $1 \leq N \leq 100$
- $1 \leq B \leq 5000$
- The cost of each balloon p_i is in the range $1 \leq p_i \leq 500$

Output

For each test case print in one line the maximum number X of teams that Félix can allow in the contest.

Example

Input	Output
3 100 1 2 3 2 1000 1 2 0 0	16 333

Problem C. Cable Way

Source file name:	`C.c, C.cpp, C.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas - CUCEI Guadalajara`

Over the mountains there is a system of basements for a secret research facility. There are many researchers that like to do all their work because it is a quiet place and also the weather is very nice since the mountains create a shadow that covers the sunlight.

A big problem is that there are no roads and helicopters can not get into the mountains, this is why they are planning to build a net of cableways to travel faster between the facility.

The terrain on the facility consists of a series of mountains that vary on height and distance. All the facilities are aligned on a line and there are also one on the edges, those two are the tallest always. Each facility is connected to at least another two facilities. The plan is to connect each one to the next facility that is taller to the left and the next taller to the right. Also, since the two at the edges are the tallest, those don't have a facility which needs to be connected directly (just have connections from lower facilities).

Given the list of facilities, the distance in meters to the leftmost and height of each facility, can you calculate the number of facilities to which each facility is connected and the distance that each cable will have?

Input

The input begins with a line with a single number N that tells how many facilities are there excluding the two at the edges. Each of the next $N + 2$ lines contains two numbers x_i and h_i, where x_i is the distance to the lsftmost facility and h_i is the height of the facility. The first facility is always located at $x = 0$ and the first and last are the same height and also the tallest. There are no two facilities at the same distance and facilities are given in increasing order of x_i in the input.

- $1 \leq N \leq 5 * 10^5$
- $1 \leq B \leq 10^8$

Output

For each test case you must print N lines with 4 numbers each, l_i, r_i, d_l, d_r, where l_i and r_i are the indexes of the facilities to which the facility i is connected to the left and to the right respectively (the first facility has index 0). d_l and d_r are the lengths of the cables that connect the facilites rounded to 4 decimas.

Example

Input	Output
4 0 10 2 8 4 2 6 2 8 8	0 5 2.8284 8.2462 1 4 6.3246 7.2111 1 4 7.2111 6.3246 0 5 8.2462 2.8284

Problem D. Dios Primes

Source file name:	`D.c, D.cpp, D.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Mario Diaz - Queen's University & Juan Pablo Marín`

In number theory, a left-truncatable prime is a primer number which, in a given base, contains no 0, and if the leading ("left") digit is successively removed, then all resulting numbers are prime. For example, 9137 is a left-truncatable prime, since $9137, 137, 37, 7$ are all prime. For this problem, decimal representation is assumed.

A Dios prime is a left-truncatable prime such that the number obtained by reversing its digits is also left-truncatable. The number 3467 is a Dios prime, since 3467 is left-truncatable and 7643 is also left-truncatable. Given two numbers a and b, your task is to find the number of Dios primes in the range $[a, b]$.

Input

The first input line contains a number T, the number of test cases. Each of the following T lines contains two numbers a and b separated by a space.

- $1 \leq T \leq 10^5$
- $1 \leq a \leq b \leq 10^9$

Output

For each test case you must print the number of Dios primes in the range $[a, b]$.

Example

Input	Output
4 1 10 11 100 3467 3467 7643 7643	4 2 1 1

Explanation

Side Remark. In order to dial *DIOS in a telephone, one has to dial *3467. That's how the Dios primes got their name.

Problem E. Endless Sum

Source file name:	`E.c, E.cpp, E.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI Guadalajara`

You are given a list of N positive integer numbers and then proceed to sum each of them in the order the numbers were given, once you reach the end of the list you will go to the first number and proceed to sum the numbers again.

At each step after you sum each number you will have a total sum S, can you determine if at any point the sum S will be a multiple of the number K ?

Input

The first line of input contains a number T, the number of test cases. Followed by T test cases, each test case contains in the first line the numbers N and K separated by a space, followed by a line with N numbers separated by a space, the list you will take to sum the numbers.

- $2 \leq T \leq 100$
- $1 \leq N < 10^5$
- $1 \leq K \leq 10^9$
- Each number A_i will be in the range : $1 \leq A_i \leq 100$

Output

For each test case you must print a line with the string "Yes." if in some point the sum will be a multiple of K, print "No." otherwise.

Example

Input	Output

`3`
`1 5`
`1`
`3 11`
`1 5 10`
	`Yes.`
	`Yes.`

Explanation

In the first case, after the first number is summed 5 times you will get a multiple of 5.

For the second case, Once you summed all the numbers your total sum is 16, you sum the first number again and the total sum is 17, next you sum the second number and the total sum is 22 which is a multiple of 11.

Problem F. Funny String

Source file name:	`F.c, F.cpp, F.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI Guadalajara`

A string is traditionally seen as a sequence of characters. Based on this definition of string, different types of strings can be defined, sorted strings for example are strings where for any consecutive pair of indexes i and j that you take in the string such that $i+1=j$ when comparing the numeric values of the characters on those possitions S_i and S_j then $S_i \leq S_j$.

Let's now define a funny String. Funny strings are similar to a sorted string, just that they have a Funny factor F such that for any consecutive pair of indexes i and j that you take in the string such that $i+1=j$ when comparing the numeric values of the characters on those possitions S_i and S_j then $S_i + F \leq S_j$.

Given a string S of lowercase characters. Can you determine what is the longest funny string that you can get by removing some characters from S ? For the input characters you can assume that the next character after "z" is "a", in such way if $S_i = z$ then $S_i + 1 = a$, $S_i + 2 = b$ and so on.

Input

The first line of input contains a number T, the number of test cases. Followed by T test cases, each test case contains a single line with the string S and the Funny factor F separated by a space.

- $2 \leq T \leq 100$
- $1 \leq F \leq 26$
- $1 \leq |S| \leq 10^3$

Output

For each test case you must print a line with a single integer. The length of the longest funny string that can be created removing some characters from S.

Example

Input	Output

`2`
`abcdefghijklmnopqrstuvwxyz 1`
`zabd 2`
	`26`
	`3`

Explanation

In the first case the given string is a funny string itself, therefore no characters should be removed.

In the second case you can remove the second character and the new string is a funny string with size $|S| - 1$.

Problem G. General pump system

Source file name:	`G.c, G.cpp, G.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Félix Arreola - CUCEI Guadalajara`

The mexican company ACM (Agua Contaminada con Microorganismos) uses a complex water pump system to clean water for the city. The company needs to replace a lot of old water pumps that have failures, they have removed the failing pumps from the map and left only the pumps that did not require a replacement.

Once the replacements were ready ACM found they did not know the place for each of the replacements as they missed to save the original map, that's why they need your help to find the way the pumps should be placed in the map.

The map is a square of size N, each cell can have maximum 1 pump, the pump system should follow some rules that should be followed carefully in order to avoid having problems with the resultant pump system:

There are 4 types of pumps (numbered from 1 to 4). A pump with number 1 needs to be connected with exactly 1 pump. A pump with number 3, needs to be connected with exactly 3 pumps, in general a pump with number n needs to be connected with exactly n pumps. Two pumps are connected if they are adjacent (share one side) in the map. If a pump does not meet this condition then the water will leak and won't be properly cleaned. For example, the following are valid configurations.

	1	
	3	1
	1	

	1	
1	3	1

	1	
1	3	
	1	

1	3	1
	1	

The pumps connections should be done carefully, if two connected pumps have the same number, they will explode. The following is an example of invalid connected pumps:

1	2	2	1

Your task is to complete the pump system, using the minimum water pumps and connecting every pump as described above. You should not move any of the working pumps (i.e the pumps given in the map).

Input

The input file contains several test cases, the first line contains an integer T the number of test casses. Following T test cases for each test case the first line contains an integer M indicating the size of the room system. The room is always a square. There are M lines following. Each line contains exactly M characters, a dot "." represents an empty space and a number (from 1 to 4) represents an already set pump. The input finishes on the end of file.

- $1 \leq T \leq 15$
- $5 \leq M \leq 8$

Output

For every test case you should output the full room, with every pump connected according to the rules. It is guaranteed that the solution exists and is unique.

Example

Input	Output
2 5 .1... ....3 .4.4. 2.42. ..1.. 6 1...2. ...4.1 3.4... ...2.1 1.2... .1...2	.1..1 23.23 34342 2342. ..1.. 1.232. 2.3431 3243.. 2.32.1 1.2.23 .13232

Problem H. Hidden card trick

Source file name:	`H.c, H.cpp, H.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas - CUCEI Guadalajara`

There is a guy that comes to CUCEI and do some magick tricks, this is his way to make money and keep studying. A lot of people have seen his magic tricks but nobody know exactly if he really studies or not. His favorite tricks are always with a deck of cards and nobody have ever found how does he do all these sorceries.

One of the tricks that I can remember is that one day he asked a friend to choose a card without letting him see it, then he shuffles the deck and tries to guess what card my friend picked, He guessed incorrectly three times with different cards and when we thought he had failed he asked us to move one step back to give him some space, when we moved back the card was under my friend's foot. Amazing!

Another trick that he does is that he asks you to take one card and to remove it from the deck, then he takes the card and places it in some random position in the deck. After the card is placed he will take the first card and put it at the bottom of the deck, and discard the next card, he continues alternating until only one card is left. As you may guess the last card is the one that was chosen at the beginning of the trick.

Can you guess how did he do this magic trick?

Given a number N representing the size of the deck determine in which position you should place the card so it will be the last card after when doing the trick.

Input

The input consists of several test cases, one case per line. Each test case is described by a single line with a single number N.

- $1 \leq N \leq 10^{15}$

Output

One line for each test case telling the position in which the card should be placed. The input finishes with a line containing a zer, this should not be processed.

Example

Input	Output
1 7 14 8 0	1 7 13 1

Explanation

For the case for 7 cards:

- 1 goes to bottom
- 2 is discarded
- 3 goes to bottom
- 4 is discarded

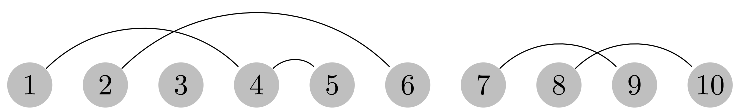
- 5 goes to bottom
- 6 is discarded
- 7 goes to bootm
- 1 is discarded
- 3 goes to bottom
- 5 is discarded
- 7 goes to bottom
- 1 is discarded
- 7 is the last card, so 7 should be the position to place the card.

Problem I. Intersecting edges

Source file name:	`I.c, I.cpp, I.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marin - CUCEI Guadalajara`

Last data structures lecture was about graphs, now you can't stop seeing a graph on everything. You say everything is a graph and maybe there is some of true on it, however your obsession with graphs has taken you to imagine some scenarios, one of those scenarios you are interesting about is the number of edges that intersect when drawing a specific type of graph.

You take a graph with N nodes and draw all the nodes on a line numbered from 1 to N. Next you will draw M edges selecting two nodes from the graph a and b and draw the edge starting in a and finishing in b drawing it above all nodes between a and b. The following is an example of the graph where $N = 10$.



As you can see this graph has two intersections.

- The edge that goes from 1 to 4 intersects with the edge that goes from 2 to 6.
- The edge that goes from 8 to 10 intersects with the edge that goes from 7 to 9.

Given a specific graph, can you determine how many intersections will be in the drawing?

Input

The first line of input contains a number T, the number of test cases. Each test case will begin with two numbers separated by a space N and M the following M lines of the test case will contain two integers separated by a space a and b representing the edges that will be drawn. No pair of edges will start or end on the same node.

- $1 \leq N \leq 10^9$
- $1 \leq M \leq 10^5$
- $1 \leq a < b \leq N$

Output

For each test case your program must output a line with a single number, the number of intersections that the drawing will have.

Example

Input	Output
2 10 5 4 5 1 4 2 6 7 9 8 10 10 5 1 2 2 3 3 4 4 5 5 6	2 0

Explanation

The input contains two test cases. The first test case is the case for the image shown above. In the second test case even when some pairs share a node no edges intersect.

Problem J. Jumping Frog

Source file name:	`J.c, J.cpp, J.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marin - CUCEI Guadalajara`

On Chapala lake there are N stones numbered from 1 to N, there is a male frog resting on stone X and a female frog resting on stone Y. Some research has shown that the frogs in Chapala are smarter than an average frog, and these studies have shown they have knowledge on some of the math areas. A female frog can only be impressed by a male frog that show knowledge on the most math areas in the lake, this is why male frogs in Chapala lake do a strange ritual to impress females, this ritual is as follows:

- To show numbers knowledge the male frog sits on a stone and selects a prime number p and jump p stones at a time until it reaches the female frog, then the frog will return to the stone where it started jumping and select another prime number that allows it to reach the female,if such number does not exist the male stops.
- To show sets theory knowledge, the male frog will take a subset of the stones that are in the range that goes from X to $Y-1$ and show his number knowledge from each of the stones in the subset.
- To show combinatorics knowledge, the male frog will do the previous for all the different subsets it can take in the range X to $Y-1$.

As you can see the male frog will reach the female several times during the ritual.

Can you help researchers to determine the number of times the male frog will reach the female knowing the stones where both frogs are resting before the male starts the ritual?

Input

The input contains several test cases, each test case contains three numbers separated by a space N, X, Y. The end of input is given in a case where all values are equal to 0, and should not be processed

- $2 \leq N \leq 10^6$
- $1 \leq X < Y \leq N$

Output

For each test case your program must output a line with a single number, the number of times the male frog will reach the female frog. As this number can be large print it modulo $10^9 + 7$

Example

Input	Output
`5 1 3` `0 0 0`	`2`

Explanation

There is only one test case in the input. There are 5 stones in the lake, the male frog rests at stone 1 and the female frog rests at stone 3. There are a total of 3 different subsets the male frog will select to impress the female frog : $\{\{1\}, \{2\}, \{1, 2\}\}$. In the first set the male frog can reach the female frog jumping from stone 1 to 3 using the prime number 2, in the second subset the male frog can not select a prime to reach the female frog, in the third subset the male frog can reach the female frog only from stone 1 using the prime number 2. In total the male frog will reach only 2 times the female frog.

Problem K. King of music

Source file name:	`K.c, K.cpp, K.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marin - CUCEI Guadalajara`

The king of music has lost his composition validation device. The composition validation device shows the king of music if a given composition will be a huge hit or not, he has explained to us how a composition is known to be a hit. A composition can be seen as an arrangement of the intensity of the tones on the time, there are some "crescendos" and "decrescendos". A "crescendo" is a gradual increase in the sound intensity while a "decrescendo" is a gradual decrease in the sound intensity over some time lapse in the composition.

The secret of the king of music is that he always composes his music in such a way that if you take the intensity at any given time t then none of the intensities in the lapse from time $t - W$ to $t - 1$ are greater or lower to K units when compared to the intensity in t. This method is why he earned the "King" title, however, since his composition validation device is lost he is unable to determine if a random composition will be a hit or not, that's why he asks you to write a program to replace his device, your program should validate a given composition and print if it will be a hit or not, i.e it complies with the rule mentioned above.

Input

The input contains several test cases, each test case starts with three numbers separated by a space N, W, K. Representing the number of intensities the composition have, and the values of W and K as mentioned above. The next line will contain the N intensities separated by a space. The end of input is given in a case where the values for N, W and K are equal to 0, and should not be processed

- $1 \leq N \leq 10^6$
- $1 \leq W \leq N$
- $1 \leq K \leq 10^6$
- Intensity in each time C_i will be in the range : $1 \leq C_i \leq 10^6$

Output

For each test case your program must output a line with the string "Yes." if the given composition complies with the king of music rules, you must print "No." otherwise.

Example

Input	Output

`5 1 5`
`1 6 7 6 7`
`5 2 5`
`1 6 7 6 7`
`0 0 0`
	`Yes.`
	`No.`

Explanation

In the first test case you must look only to the previous intensity and none of the intensities in the lapse for all possible lapses should be less than or more than 5 units. All time lapses comply.

In the second test case you must look to the two previous intensity for any intensity and none of the intensities in the lapse for all posible lapses should be less than or more than 5 units. The lapse 1 6 7 does not comply as 1 is more than 5 units less than 7.

Problem L. Lowest Terms Fractions

Source file name:	`L.c, L.cpp, L.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Gilberto Vargas - CUCEI Guadalajara & Juan Pablo Marín`

Mr. Homft's today lecture is about fractions. Fractions are the numbers that describes the *ratio* between values. For example, the *Golden Ratio* is described by the fraction $\frac{fib(n+1)}{fib(n)}$, which tends to be 1.61803. Note that $fib(n)$ is the n^{th} term of fibonnaci's sequence.

Mr. Homft explained also the types of fraction there are. The proper fractions are the ones of the form $\frac{a}{b}$ where $a < b$, for example $\frac{3}{4}$, $\frac{5}{7}$ and $\frac{1}{8}$ are proper fractions. Improper fractions are the ones that are not proper fractions or the ones where $a \geq b$, like $\frac{6}{5}$ or $\frac{8}{7}$.

After explaining all that, he mentioned that a good mathematician always simplifies the fractions he writes, for example, $\frac{4}{8}$ should never be written, because it can be simplified into $\frac{1}{2}$, these fractions are known as irreducible fractions or lowest terms fractions.

After all that was explained he wrote the homework:

Given a number n, how many fractions of the form $\frac{x}{n}$ are there which cannot be simplified, and what is their sum?

Input

The input starts with a line containing a single integer T the number of test cases. Each of the next t lines contains a single number n.

- $1 \leq T \leq 10^4$
- $2 \leq N \leq 10^9$

Output

Your program should output a line per test case with two numbers separated by a space, the first one being the number of lowest terms fractions of the form $\frac{x}{n}$, the second one the sum of these fractions rounded to 4 digits.

Example

Input	Output

`4`
`2`
`3`
`4`
`5`
	`1 0.5000`
	`2 1.0000`
	`2 1.0000`
	`4 2.0000`

Explanation

In the first test case there is only one lowest term fraction on the form $\frac{x}{n}$ being $\frac{1}{2}$, the sum is then $\frac{1}{2} = 0.5000$. For the last test case there are 4 lowest terms fractions: $\frac{1}{5}, \frac{2}{5}, \frac{3}{5}$ $\frac{4}{5}$ and their sum is $\frac{1}{5} + \frac{2}{5} + \frac{3}{5} + \frac{4}{5} = \frac{10}{5} = 2$

2017

acm icpc

icpc.foundation

Gran Premio de México
y Centroamérica

Coordinado por

ITESO, Universidad
Jesuita de Guadalajara

TERCERA FECHA

Problem A. Advanced Genetics

Source file name:	A.c, A.cpp, A.java, A.py
Input:	`Standard`
Output:	`Standard`

Genetics is an interesting biology field which also intersects with other areas of science and is strongly linked with the study of information systems, genetics studies genes, genetic variation, and heredity in living organisms.

Genes contain the information that builds DNA, DNA consists of a chain made from four types of nucleotide subunits : adenine, cytosine, guanine and thymine.

People from Triplonia have found that their DNA string is formed of several gene strings that are merged after a special gene is added to mix. This special gene is called the "Merging factor" . A gene string is a way to represent genes, it is a string that only contains the characters 'A', 'C', 'G' and 'T' each of these characters represents one of the nucleotide subunits that create genes and DNA.

Triplonia genetics researchers have collected a set S of N gene strings and are ready to apply the merging factor M, however, since the merging of the strings occurs after adding the merging factor to the genes mix they want to know how many of the gene strings will be merged so that they are ready to get more samples after this test. Getting genes is an expensive task for them, that's why they have explained the process and are looking for someone who can tell them the number of gene strings that will be merged. Can you help them?

The merging process is as follows :

1. Set $R = M$
2. If suffix of length K of R and the prefix of length K of a string S_i in S are anagrams, then S_i is appended to R.
3. Repeat step 2 with $S = S_{i+1}, S_{i+2}, ..., S_N$ until no more strings can be merged.

One more important thing to know is, if there are more than one way to merge the strings, the resulting DNA will always be the one that merges the most number of strings.

Given N, K, M, and S. Can you help scientists to know, what will be the number of strings merged to create the DNA?

Input

The input starts with a single line containing a single number T . The number of tests cases. Each test case will start with a line containing two integer numbers N and K, the number of genetic strings in the set and the length of the strings merging factor. The next line contains a string M the merging factor used to start the merging process. Each of the next N lines contains a string in the set to merge.

- $2 \leq N \leq 10^3$
- $1 \leq K \leq 10^3$
- All strings will be at least K in length and will not exceed 10^3 characters

Output

For each test case print a single line the maximum number of gene strings that can be merged with process mentioned above.

Example

Input	Output
1 5 2 AC ACTGT TGACC CCGCA CCGGG GGACTG	4

Explanation

In the test case, there are 5 gene strings, the prefixes and suffixes should be of length 2 and the merging factor is AC. The merging factor can be merged with ACTGT (the first gene string). Then it can be merged with TGACC, which can be merged with CCGCA which can not be merged with another string having a total of 3 strings merged, however, if instead of merging CCGCA the string CCGGG is merged then GGACTG can be merged giving a total of 4 strings merged. Since there are no way of merging more than 4 strings then the maximum number of gene strings that can be merged is 4.

Problem B. Banking

Source file name:	B.c, B.cpp, B.java, B.py
Input:	`Standard`
Output:	`Standard`

You are excited to work on a new project related to banking and bank accounts. The ACM bank have created a new type of account, the save much and spend a lot account, the idea of such accounts is to help people to save money and after they have saved to only spend.

The save much and spend a lot accounts have the following two rules:

- If you have not spent any money, then you can add money or spend money.
- If you have spent money, you can only spend money

The project you are working on the bank is to determine and fix some errors that were found on the database. In some accounts you can see how money is spent and then added again which violates the rules for the save much and spend a lot accounts, it was found that some issues on the database were adding information to accounts that did not make some movements so your task is given the amount of money an account had over time determine what are the maximum number of movements that comply with the save much and spend a lot account rules in that specific account.

Input

The input consist of several test cases. The first line of input contains a single number T, the number of test cases followed by T test cases. Each test case begins with a line containing a single number N the number of movements stored in the database for an account. The next line contains N numbers separated by a space, the amount of money the account had after the registered movement i.

- $1 \leq N \leq 1000$
- The amount of money in the account for any movement A_i will always be in the range $0 \leq A_i \leq 10^6$

Output

For each test case print in one line the maximum number X of movements that comply with the save much and spend a lot account rules.

Example

Input	Output
2 5 4 2 3 5 1 8 1 9 11 2 3 1 5 10	4 6

Problem C. Coming home

Source file name: C.c, C.cpp, C.java, C.py
Input: `Standard`
Output: `Standard`

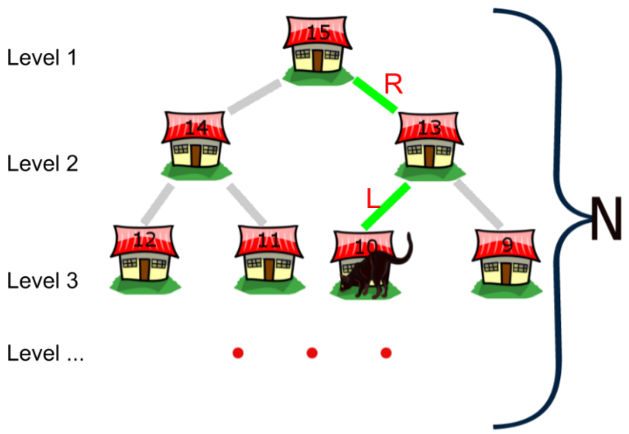
Baker is a very curious cat, this morning he went out of the neighborhood to explore the world, but, when he decided to come home he realized he was lost. Since baker is a very smart cat he remembers the number b of the house where we live.

Let me explain how the neighborhood where we live is, it is a very exclusive place, organized in N levels, top levels have more expensive houses also they are farther from the neighborhood entrance and have a lower number. To get to a house you can get to it only from a house with higher number and each house except for those on the top level are connected to two houses, one taking the path to the left and one taking the path to the right, both of them on the next level of the neighborhood. Each house except for the only house in the first level is also connected to a house with a higher number in the previous level. There is no path between any two houses in the same level k.

Houses are numbered with consecutive numbers from 1 to c where c is the total number of houses in the neighborhood. The most exclusive house on each level is located at the right, and the numbering goes from right to left on the levels starting on the higher levels to the lower levels.

The path to get Baker to the house from the entrance is unique and even when Baker is very smart he only understands the commands “Go left” (L) and “Go right” (R) .

Given the number of levels in the neighborhood and the number of the house where Baker lives, you must tell the instructions to Baker in order for him to come home.



In picture you can see Baker neighborhood when the number of levels is 4. Baker lives in the house with number 10. The instructions for Baker to get home from the neighborhood entrance are : “Go Right”, “Go Left” (RL).

Input

The input contains several test cases. Each test case contains a single line with two numbers N and b.

- $1 \leq N \leq 30$
- $1 \leq b \leq 2^N - 1$

Output

For each test case you must print a single line with the string "Instructions: " without the quotes followed by the instructions baker has to follow. 'R' should be used for a "Go Right" instruction and 'L' should be used for a "Go Left" instruction.

Example

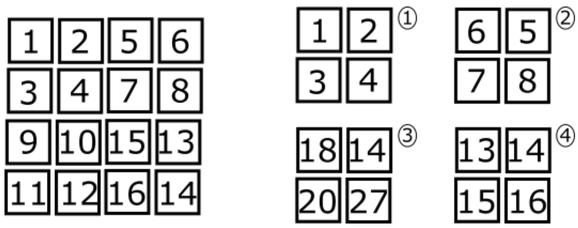
Input	Output
4 10 4 15 6 31	Instructions: RL Instructions: Instructions: LLLLR

Problem D. Divided square

Source file name: D.c, D.cpp, D.java, D.py
Input: `Standard`
Output: `Standard`

Divided square is a new online puzzle. In this puzzle you have a square with size N where N is an even number divided in N x N cells, each cell has a number. You are provided with a set of K tiles with size 2 x 2, these tiles have also 2 x 2 cells, and each cell has a number, each of these tiles have also a number P_i of points that you won if you use the tile in the game.

In Divided square you have to select a subset of the K tiles in such way that you can fill the square with the tiles, a tile can be added in a place of the square if the numbers in the tile and in the square matches the position where you want to place the tiles. You can rotate tiles in order to make the tile match the numbers in the position.



The image shows a 4 x 4 square. And 4 tiles numbered as 1, 2, 3, and 4. The tile with the number 1 matches the square in the position $(1, 1)$. The tile with the number 2 does not match the square in any position. The tile with the number 3 does not match the square in any position. The tile with the number 4 matches the square at position $(3, 3)$ after rotating it one time clockwise.

As you can guess, your task is to determine :

1. If you can fill the square with the given tiles

2. What is the maximum number of points you can get after filling the square with the tiles

Input

The input consists of several test cases. Each test case starts with two numbers N the size of the square and K the number of tiles. The next 3*K lines describe the tiles in the game, the description of a tile is given in two lines with two numbers that represents the numbers that contains the 2 x 2 square of the tile. The third line in each tile description represents P_i the number of points you will get if using the i-th tile. Each of the next N lines contains N numbers representing the numbers in the cells of the N x N square. The input ends with a case where $N = 0$ and $K = 0$. This test case should not be processed.

- $1 \leq N \leq 500$
- $1 \leq K \leq 10^5$
- The value c for any cell will be in the range : $0 \leq c \leq 100$

Output

For each test case print a line with a single number, the maximum number of points you can get if the square can be filled with the given tiles, print -1 if the square can not be filled with the given tiles.

Example

Input	Output
4 5 1 2 3 4 1 5 6 7 8 4 9 10 11 12 3 9 10 13 14 3 13 14 15 16 5 1 2 5 6 3 4 7 8 9 10 15 13 11 12 16 14 4 4 1 2 3 4 1 5 6 7 8 4 9 10 11 12 3 9 10 13 14 3 1 2 5 6 3 4 7 8 9 10 15 13 11 12 16 14 0 0	13 -1

Explanation

In the first test case there is a 4 x 4 square and 5 tiles. You can fill the square taking the tiles number $1, 2, 3, 5$ the tile number 5 should be rotated to match in the square. The total number of points to earn is 13.

The second test case is exactly the same as the first test case but without the 5^{th} tile. The square can not be filled with the given tiles therefore the output is -1.

Problem E. Pythagorean triple

Source file name:	E.c, E.cpp, E.java, E.py
Input:	`Standard`
Output:	`Standard`

Andrew has been interested since he was a kid on some number theory topics. He has always looked for numbers with interesting properties, his friends may call them "weird". He is now a college student and is very interested in his programming class since he can apply his math skills in the programming field.

He is studying what is called Pythagorean triples, a Pythagorean triple consists of three positive integers a, b, and c, such that $a^2 + b^2 = c^2$. Such a triple is commonly written (a, b, c), and a well-known example is $(3, 4, 5)$.

Your task is given a number N to find all Pythagorean triples that exist where $a \leq b \leq c \leq N$.

Input

The input consists of several test cases, each test case contains a unique line with a single number N. The input ends with a test case where $N = 0$, this test should not be processed.

- $1 \leq N \leq 512$

Output

For each test case you must print a line with a number. The number of Pythagorean triples (a, b, c) that exist in the range $1 \leq c \leq N$

Example

Input	Output
1 10 100 0	0 2 52

Problem F. Urban Map

Source file name:	F.c, F.cpp, F.java, F.py
Input:	`Standard`
Output:	`Standard`

Urban planning in a city requires usage of geographic maps that represents some parts of the city. In each map you can see buildings that are planned to be built on that zone. In order to follow current urban law codes it is need to verify that the area of buildings is not greater than some percentage from the total geographic area.

Town planners require help to determine what is the area of the map that is not part of a building. A map is a rectangle of R rows and C columns, each cell in the map can may be built or empty. If a set of built cells surround cells that are not built then these cells are considered to be built as they are inside of a building. A building may contain or not empty cells inside.

Any cell that is not part of a building is called "outside cell", all of these cells are connected, two cells are connected if they share a side in the map. There is always at least one "outside cell" in the periphery of the map. Your task is to determine the number of "outside cells" that exist in the map.

In the figure you can see the map from the test case and it contains three buildings. The biggest has a set of 7 empty cells inside it and in takes 21 cells in total, there is also other building which does not contains empty cells inside it and takes 3 cells in total, and the third building contains two empty cells inside it and 9 cells in total. The total area of the map $8x8 = 64$ minus the area taken by buildings 33 gives the number of "outside cells" $64 - 33 = 31$

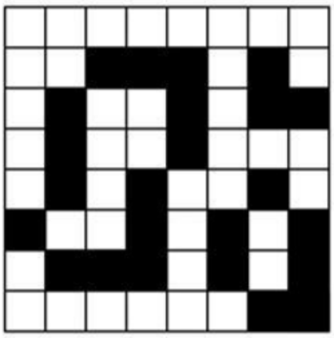


Figura 1. Ejemplo de un mapa de 8 x 8 celdas

Input

The first line of input contains a number T the number of test cases. Each test case starts with a line with a single number R the number of rows, the next line contains a single number C the number of columns. The next R lines contains C numbers each number represents if that cell in the map is built (1) or not (0) (Note that there are no spaces between the numbers as shown in the sample input).

- $2 \leq T \leq 10$
- $2 \leq R \leq 30$
- $2 \leq C \leq 30$

Output

For each test case your program should print a single number representing the number of "outside cells" that exist on the map.

Example

Input	Output
1 8 8 00000000 00111010 01001011 01001000 01010010 10010101 01110101 00000011	31

Problem G. Electric cabling

Source file name:	G.c, G.cpp, G.java, G.py
Input:	`Standard`
Output:	`Standard`

Cities for the future are being built in this age, some of their new technologies are nanotechnology electrical cabling, these new way of cabling repairs automatically any problems found on the electrical flow. The cabling is done connecting a set of N poles of several sizes in line with the new nano-electrical wires, the first pole is connected to an electrical generator which makes the electrical flow to go from the first pole to the last one.

Some testing on the new technology have found that the nanotechnology is not working properly on some of the cabling poles, project engineers have found the problem is related to the poles with the highest height, if there are two poles with the highest height then all poles between them and including the highest poles will not get proper electrical flow.

If there are more than two poles with the highest height, the electrical flow will get corrected after the second highest pole, then on the third it starts to fail again until it gets to a fourth highest pole or to the end of the electrical cabling, and so on.

If there is only one pole with the highest height, the electrical flow will fail in all the poles after the highest one.

Given the number of poles in the cabling and the height of each of these cables, you must find the number of poles that will not get a proper electrical flow.

Input

First line of input contains a number T, the number of test cases. For each test case you have a single line containing a set of S numbers separated by a space representing the height of each pole.

- $1 \leq T \leq 50$
- $1 \leq S \leq 100$

Output

For each test case you must print a single line with the number of poles that will not get proper electrical flow.

Example

Input	Output
3 100 254 128 129 254 253 467 718 920 540 920 101 130 131 130 131 130 131 130 131 130	4 3 6

Problem H. Patty's gift

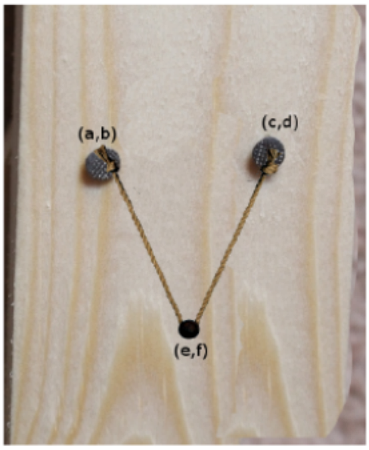
Source file name: H.c, H.cpp, H.java, H.py
Input: `Standard`
Output: `Standard`

Last Christmas, Patty was very busy embroidering a beautiful hand-made painting for Tony. When she finished, she placed a tiny hook behind the frame so that Tony could hang it on the wall.

Tony is very bad at decorating his house, but he really likes his gift so he wants it to be placed in the best place of the house. He plans to nail two nails on the wall with a rope tied to them, and he wants to have a preview of where the painting will be placed at the end, before he ruins his wall with the holes of misplaced nails.

He has drawn a Cartesian plane on the wall, has chosen two arbitrary points (say (a, b) and (c, d)) where he pretends to nail each nail, and he plans to use a rope of an arbitrary length (say L). Now... he doesn't know where the tiny hook of his gift will be located.

Please help Tony to visualize his new decoration, calculating where that hook will be. Assume that the painting is heavy enough to keep the rope tense, and the knots to tie it to the nails don't diminishes the length. Tony's world is very boring, so friction does not exist.



Input

The input consists of several test cases, each one in a single line. For each line you are to read the coordinates of the nails (a, b) and (c, d)) and the length of the rope, L each value is separated by a space.

- $-1000 \leq a, b, c, d \leq 1000$
- $0 \leq L \leq 5000$

Output

You have to calculate the coordinates (e, f) of the point where the hook will be placed, after the frame is hung on the rope. Your answer will be considered correct if the difference respect to the exact solution is less than 10^6.

Example

Input	Output
2 5 7 4 5.009	7 4
2 5 7 4 10	4.788675 0.169873
10 20 30 40 31	11.555993 18.15728

Problem I. Traffic flow

Source file name:	I.c, I.cpp, I.java, I.py
Input:	Standard
Output:	Standard

A city has n intersections and m bidirectional roads connecting pairs of intersections. Each road has a certain traffic flow capacity, measured in cars per minute. There is a path from every intersection to every other intersection along some sequence of roads. The road maintenance department is over budget and needs to close as many roads as possible without disconnecting any intersections. They want to do it in such a way that the minimum capacity among all of the remaining roads is as large as possible.

Input

The first line of input gives the number of cases, T. T test cases follow. Each one starts with a line containing n $(0 \leq n \leq 100)$ and m $(0 \leq m \leq 10000)$. The next m lines will describe the m roads, each one using 3 integers, u, v and c $(0 \leq u, v \leq n)$, $(0 \leq c \leq 1000)$. u and v are the endpoints of the road and c is its capacity.

Output

For each test case, output one line containing "Case #x: " followed by the capacity of the minimum-capacity remaining road.

Example

Input	Output
2 2 3 0 1 10 0 1 20 0 0 30 4 5 0 1 1 3 1 2 1 2 3 2 3 4 0 2 5	Case #1: 20 Case #2: 3

Problem J. Anagrams by Stack

Source file name: J.c, J.cpp, J.java, J.py
Input: Standard
Output: Standard

A school teacher wants to simplify the concept of a stack and its operations to his classroom. He wants a program to show them how anagrams can result from sequences of stack operations: 'push' and 'pop'. He'll be illustrating on 4- letter words. There are two sequences of stack operators which can convert TROT to TORT:

[
i i i i o o o o
i o i i o o i o
]

where i stands for Push and o stands for Pop. Given pairs of words, your program should produce sequences of stack operations which convert the first word to the second.

A stack is a data storage and retrieval structure permitting two operations:

Push - to insert an item and

Pop - to retrieve the most recently pushed item

We will use the symbol i (in) for push and o (out) for pop operations for an initially empty stack of characters. Given an input word, some sequences of push and pop operations are valid in that every character of the word is both pushed and popped, and furthermore, no attempt is ever made to pop the empty stack. For example, if the word FOO is input, then the sequence:

i i o i o o is valid, but

i i o

Is not (it's too short), neither is

i i o o o i

(there's an illegal pop of an empty stack)Valid sequences yield rearrangements of the letters in an input word. For example, the input word FOO and the sequence i i o i o o produce the anagram OOF. So also would the sequence i i i o o o. You are to write a program to input pairs of words and output all the valid sequences of i and o which will produce the second member of each pair from the first.

Input

The input file will consist of several pairs of input lines. The first line of the file will include a single integer, indicating the number of pairs of input lines that will follow. The first line of each pair of input lines is to be considered as a 4-letter source word .The second line is the 4-letter target word.

Output

For each input pair, your program should produce a sorted list of valid sequences of i and o which produce the target word from the source word. Each list should be delimited by

[
]

on a separate line. The sequences should be printed in "dictionary order". Within each sequence, each i and o should be followed by a single space. Each sequence should be on a separate line.

Example

Input	Output
3 mada adam long nice eric rice	Output for mada adam [i i i i o o o o i i o i o i o o] Output for long nice [ ] Output for eric rice [i i o i o i o o]

Problem K. Graph coloring

Source file name:	K.c, K.cpp, K.java, K.py
Input:	`Standard`
Output:	`Standard`

As you learned in your Algorithms class(es), the 4 colors theorem only applies to planar graphs (that is, graphs that can be drawn without edges intersecting each other, except on their vertices).

However, you can always color a graph according to the rules of the theorem (no adjacent nodes can have the same color) with a number of colors $C \leq N$, the number of nodes (ultimately you can color each node with a different color).

On this problem, you will strive to find the minimum number of colors (min C) that you need to color all nodes from a directed graph following those same rules. Remember that in a directed graph an edge has direction, so if an edge connects A to B, that does not imply that B connects to A.

Input

Each test case input will be separated by a blank line and the end of the input is indicated by the EOF.

Each test case will be formatted as follows: On the first line a single integer $0 \leq N \leq 10$ indicates the number of nodes in the graph. The following N lines describe the information of the edges. The first integer $0 \leq E \leq N$ on line x indicate the number of edges connecting node x to another node. The rest E integers indicate the node number that each edge connects to. Nodes are enumerated from 1 to N.

Output

For each test case all you have to do is print a line with the minimum number of colors required to color the graph with the rules indicated above.

Example

Input	Output
4 2 2 3 2 1 3 3 1 2 4 1 3 5 4 2 3 4 5 4 1 3 4 5 4 1 2 4 5 4 1 2 3 5 4 1 2 3 4	3 5

Problem L. Lemon trees

Source file name:	L.c, L.cpp, L.java, L.py
Input:	`Standard`
Output:	`Standard`

Lemon is a very popular citric fruit used to make a lot of beverages and giving that very known sour taste to almost every meal where it is added.

As you can imagine as most people uses lemons there are also a lot of people which works harvesting lemons to provide the amount of lemon that is needed in the region.

Joseph is owner of N lemon trees in a line of land, each of these trees have an amount of T_i lemons that are ready to be harvested by K people. As there is a high demand for lemon on the region this summer Joseph needs all the lemons to be harvested but he is also conscious that people may get tired of all the harvesting work, that's why he has decided that each person will work with a set of contiguous lemon trees.

Once a person starts harvesting the tree i the person will collect all the T_i lemons the tree has, the total work a person does is measured by the number of lemons the person collected, this is, the sum of lemons for each of the lemon trees the person harvested.

Joseph needs your help to find a way to select the segments of trees each person will work, in such a way that the person who collects more lemons is the minimum possible.

Input

The input contains several test cases. Each test case is described by two lines: the first line of each test case contains two numbers N the amount of lemon trees and K the amount of people working on the lemon harvesting, the second line contains N numbers separated by a space the amount of lemons each lemon tree has.

- $1 \leq N, K \leq 10^5$
- $1 \leq T_i \leq 10^6$

Output

For each test case you should print a line with a single number, the number of lemons collected by the person that collected more lemons based on the given conditions.

Example

Input	Output
6 3 2 2 2 2 2 2 6 3 6 4 3 2 3 1	4 7

Explanation

In the first test case the work can be distributed evenly to the 3 workers, two lemon trees for each and the one who collects more in this case is 4, any other arrangement will have a worker with more than 4 lemons collected.

The second test case is a little more tricky, the first worker will harvest only the first lemon tree, the second worker harvests the second and third lemon tree and the third worker harvests the last three lemon trees, the one that collects more lemons is the second worker, who will collect 7 lemons. Any other arrangement requires a worker to collect more than 7 lemons.

2017

acm icpc

icpc.foundation

Gran Premio de México y Centroamérica

Coordinado por

ITESO, Universidad Jesuita de Guadalajara

REPECHAJE

Problem A. Medium Size Summations

Source file name:	A.c, A.cpp, A.java, A.py
Input:	`Standard`
Output:	`Standard`

R2-D2 (our well known friendly robot) needs to perform some operations quickly to save his space ship. These operations require computing long summations and a division. Moreover, he needs to find the exact solution and he is required to present a report with the results. For that, he needs to simplify his solution as much as possible.

We assume that there is an array available $(X1,\ X2,\ \ldots\ )$ of 99999999 elements.

The array has the peculiar property that the average of the first K numbers is equal to the average of the index K and the number 1.

R2-D2 needs to do the following: Given a natural number N less than 99999999, his assignment is to compute the function :

$F(N) = F1(N)/F2(N)$ where:

$F1(N) = N \times \left(\sum_{1\leq k\leq N} \frac{k^4}{X_k}\right)$

$F2(N) = \left(\sum_{1\leq k\leq N} \frac{k^3}{X_k}\right) \times \left(\sum_{1\leq k\leq N} \frac{k^2}{X_k}\right)$

That is, $F1(N) = N \times \left(\frac{1}{X_1} + \frac{16}{X_2} + \ldots + \frac{N^4}{X_N}\right)$ and

$F2(N) = \left(\frac{1}{X_1} + \frac{8}{X_2} + \ldots + \frac{N^3}{X_N}\right) \times \left(\frac{1}{X_1} + \frac{4}{X_2} + \ldots + \frac{N^2}{X_N}\right)$

Since R2-D2 needs an exact solution, we ask him to report the following: The solution needs to be given as a pair of relative prime numbers a,b such that $F(N) = a/b$ if the solution is not an exact integer. Otherwise just give the exact integer. The numbers processed by R2-D2 were of eight digits (99999999). Remember that R2-D2 was built long long time ago. His circuits are not that fast but he is clever. R2-D2 was able to perform one of these operations in less than one second. Can you do this assignment as fast as R2-D2 did it?

Input

You will receive an input line with natural numbers, one per line. Each number is less than 99999999. You will receive no more than 20 numbers.

Output

You need to give a sequence of lines each one with the solution of the corresponding input case. The solution is either a pair of natural numbers separated by the symbol / representing the pair a,b mentioned above (when the division is not exact) or just one natural number (when the division is exact). Notice that these numbers could require more than 8 digits.

Example

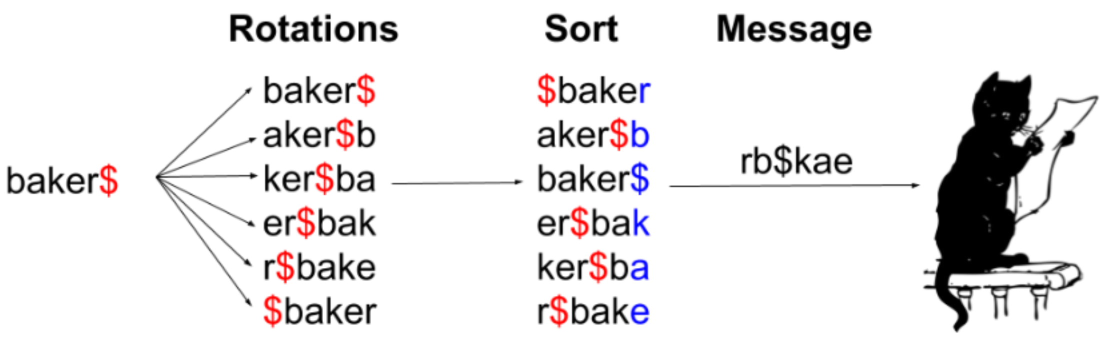
Input	Output
1 2	1 6/5

Problem B. Baker$

Source file name:	B.c, B.cpp, B.java, B.py
Input:	Standard
Output:	Standard

Baker and Iris are really good friends. Last month Iris moved out of the city and now their only way to communicate is using letters. As both Baker and Iris are cats, their each of their letters contains a single string of N characters. In order to create a more secure way of communication, Iris decided that she will encrypt the messages that sends to Baker using the following procedure:

First add the character $ at the end of the string, next she will create all rotations of the string in lexicographical order.Consider the character $ is the lowest lexicographically. The encrypted string is made taking the last character of each rotation.



Your task is to help Baker to decrypt the letter Iris has sent.

Input

The input consists of several test cases. Each test case consists of a line with a string S that contains only lower case characters and the $ symbol.

- $1 \leq |S| \leq 1000$

Output

For each test case print in one line the decrypted message.

Example

Input	Output
rb$kae annb$aa	baker banana

Problem C. Binomial Powers

Source file name:	C.c, C.cpp, C.java, C.py
Input:	`Standard`
Output:	`Standard`

In algebra, one of the classical examples of algebraic expansions is the binomial power $(x + y)^n$, where $n \geq 0$ is an integer.

Here are a couple examples of these expansions:

$(x + y)^2 = x^2 + 2xy + y^2$

$(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$

As you can see, the number of terms in the expansion is always $n + 1$ and the exponents on x decrease from n to 0 while the exponents on y increase from 0 to n along these terms.

The only thing that is not so obvious is the values of each term's coefficient. An interesting result from algebra is that coefficients are very easy to calculate.

If you sort the terms in the ordering described above (x exponents decreasing), the value of the k-th term's coefficient is ${}_nC_k$, which is usually called "n choose k" and is defined as follows:

${}_nC_k = \frac{n!}{(k!(n-k)!)}$

Fortunately, a recursive definition exists for the binomial coefficients and is calculated as follows:

For n, $k > 0$; ${}_nC_k =_{(n-1)} C_{(k-1)} +_{(n-1)} C_k$

For $n \geq 0$; ${}_nC_0 = 1$; ${}_nC_n = 1$

Your job here is to expand a bunch of binomial powers as explained.

Input

The input starts with a line with a single integer T, followed by T test cases. Each test case is written in a single line and contains a single integer n.

- $1 \leq T \leq 100$
- $0 \leq n \leq 100$

Output

For each test case you need to calculate the expansion of $(x + y)$.

The output must be written as in the sample output, with exponents written after a '^' symbol and no spaces between any characters. Terms must be separated by a '+' character. Also remember that the terms must be sorted as explained before.

If a variable in a term has exponent 1, you must not write the exponent nor the '^' symbol. If the exponent is 0 you must also not write the variable itself. Coefficient should only be written if it is different from 1, or if both variables have exponents 0.

Examples:

- $3x^2y^0$ should be printed as 3x^2
- $1x^3y^1$ should be printed as x^3y
- $2x^0y^2$ should be printed as 2y^2

- $1x^0y^0$ should be printed as 1

Example

Input	Output
2 2 3	x^2+2xy+y^2 x^3+3x^2y+3xy^2+y^3

Problem D. Digital roots

Source file name:	D.c, D.cpp, D.java, D.py
Input:	`Standard`
Output:	`Standard`

The digital root of a positive integer is found by summing the digits of the integer. If the resulting value is a single digit then that digit is the digital root. If the resulting value contains two or more digits, those digits are summed and the process is repeated. This is continued as long as necessary to obtain a single digit.

For example, consider the positive integer 24. Adding the 2 and the 4 yields a value of 6. Since 6 is a single digit, 6 is the digital root of 24. Now consider the positive integer 39. Adding the 3 and the 9 yields 12. Since 12 is not a single digit, the process must be repeated. Adding the 1 and the 2 yields 3, a single digit and also the digital root of 39.

Input

The input file contains several test cases. Each test case contains a line with a single positive integer number N. The end of the input will be in a case where $N = 0$, this case should not be processed.

- $1 \leq N < 2^{64}$

Output

For each test case in the input, print a single line with the digital root of N.

Example

Input	Output
24 39 0	6 3

Problem E. Keeping thins in order

Source file name:	E.c, E.cpp, E.java, E.py
Input:	`Standard`
Output:	`Standard`

In this problem, you are given a series of lists containing words and numbers. Your task consists on sorting them so that the words are in alphabetical order, and the numbers in numerical order. There is one catch though: if the k-th element of the list is a number, it must remain a number in the sorted list.

Input

The input will contain multiple lists, one per line, there will be no more than 10 lists in the input. Each element will be separated by a comma followed by a space, and the list will be terminated by a period. No list will contain more than 100 elements. The input will be terminated by a line containing only a period.

Output

For each list in the input, output the sorted list, separating each element of the list with a comma followed by a space, and ending the list with a period.

Example

Input

```
0.
banana, strawberry, OrAnGe.
Banana, StRaWbErRy, orange.
10, 8, 6, 4, 2, 0.
x, 30, -20, z, 1000, 1, Y.
50, 7, kitten, puppy, 2, orangutan, 52, -100, bird, worm, 7, beetle.
.
```

Output

```
0.
banana, OrAnGe, strawberry.
Banana, orange, StRaWbErRy.
0, 2, 4, 6, 8, 10.
x, -20, 1, Y, 30, 1000, z.
-100, 2, beetle, bird, 7, kitten, 7, 50, orangutan, puppy, 52, worm.
```

Problem F. Teleporters

Source file name:	F.c, F.cpp, F.java, F.py
Input:	`Standard`
Output:	`Standard`

Mankind has evolved into a pretty awesome state. We have flying cars. We now live in planets other than Earth. But many people's favorite invention is the teleporter. Teleporters are way too expensive to be owned by every single planet. However, there are some rich planets which do have them.

The problem with the current teleporting technology is that they only work in pairs. In other words, one teleporter can only teleport people to ONE specified exit. They are bidirectional, so an "entry" teleporter can also work as an "exit" teleporter. Traveling between teleporters takes no time.

Your job as a travel planner is to come up with the best route between planets. The best route is the one that takes the less amount of time. Obviously, you can use teleporters to do this.

Input

First you will be given an integer N, the number of test cases. N maps, for which you have to calculate best routes, will follow.

For each of the N maps, the first line contains two integers, $1 \le V \le 50$, and $T \ge 0$, where V is the number of planets in the system, and T is the number of pairs of teleporters.

After these 2 numbers, V lines will follow. The first integer in these lines represents a planet, and the next pairs of numbers represent planets which can be reached from the current one, as well as the time it takes to reach them. Note that even if there is a path from planet A to planet B, that does not necessarily mean there is a path from B to A. The numbering of the planets starts with 0.

After that, there is a line with $T * 2$ integers, which represent the teleporters. These should be read pairwise, that is, if the line looks like this:

0 1 2 3

that means there is a teleporter pair between planets 0 and 1, and another pair between planets 2 and 3. If T is zero, then this line is omitted.

Finally, there is a line with two integers, the first represents the starting planet, and the second the goal.

Output

If there is a path from the starting planet to the goal planet, print one line with the format "Case #X: time" (X starts in 1), followed by a line containing all the planets visited, in order. Each of these planets will be separated by a single whitespace.

In case there exists no path, print "Case #X: Path does not exist", and omit the next line.

Note that the minimum path will always be unique.

Also, teleporter pairs can only be used once. So, if you already used a teleporter going from planet A to B, then you cannot use that same teleporter again, not even if going from B to A.

Example

Input	Output
3 4 2 0 1 10 1 2 10 2 3 10 3 0 10 3 2 2 0 1 0 3 0 0 1 10 2 20 1 2 1 2 0 2 3 0 0 1 10 2 20 1 2 1 2 1 2	Case #1: 10 1 2 0 Case #2: 20 0 2 Case #3: Path does not exist

Problem G. Prime gap

Source file name:	G.c, G.cpp, G.java, G.py
Input:	`Standard`
Output:	`Standard`

A prime gap is the difference between two successive prime numbers. The n-th prime gap, denoted $g(n)$ is the difference between the $(n+1)$-th and the n-th prime numbers, i.e.

$g(n) = p_{n+1} - p_n$

For the first five prime numbers (2,3,5,7,11) We have $g(1) = 1$, $g(2) = 2$ and $g(4) = 4$ the sequence of prime gaps has been extensively studied.

The first 30 prime gaps are :

$1, 2, 2, 4, 2, 4, 2, 4, 6, 2, 6, 4, 2, 4, 6, 6, 2, 6, 4, 2, 6, 4, 6, 8, 4, 2, 4, 2, 4, 14$

In this problem your task is to provide the maximum prime gap between two arbitrary numbers a and b.

Input

The input consist in several cases no more than 100, each case consist of two positive integers a and b, The cases will end with a line with the integers 0 0.

- $1 \leq a, b \leq 1300000$

Output

For each case you must print the maximum prime gap inside the range defined by the two positive integers inclusive, or NULL if such gap does not exist.

Example

Input	Output

`1 1`
`6 12`
`1 30`
`0 0`
	`NULL`
	`4`
	`6`

Problem H. In a hurry

Source file name: H.c, H.cpp, H.java, H.py
Input: `Standard`
Output: `Standard`

David hates to wait at stop signs, yield signs and traffic signals while driving. To minimize this aggravation, he has prepared maps of the various regions in which he frequently drives, and measured the average delay (in seconds) at each of the various intersections in these regions. He wants to find the routes between specified points in these regions which minimize his delay at intersections (regardless of the total distance he has to drive to avoid delays), and has enlisted your assistance in this effort.

Input

For each region, David provides you with a map. The map data first identifies some number of intersections, NI. The regions never include more than 10 intersections. The intersections in each region are numbered sequentially, starting with the number one (1). For each intersection, in turn, the input then specifies the number of streets leading away from the intersection, and for each such street, the number of the intersection to which the street leads, and the average delay, in seconds, that David encounters at that intersection. Following the data for the last intersection in a region there appear the numbers associated with the intersections where David wants to start and end his drive. The entire input consists of a sequence of maps, followed by the single integer zero (0). Note: between each (node number, cost) pair there are 3 white spaces.

Output

For each region, in order, print a single line of output which contains the region number (they, too, are sequentially numbered, starting with 1), a list of the intersection numbers David will encounter in the route with minimum average delay, and the average number of seconds he will be delayed while travelling this route. A suitable format is shown in the example below.

Notes:

1. There will always be a unique route with the minimum average delay in each region.
2. A street from intersection I to intersection J is one-way. To represent a two-way street from I to J, the map must also include a route from intersection J to intersection I.
3. There will never be more than one route directly from intersection I to intersection J.

Example

Input
5 2 3 3 4 6 3 1 2 3 7 5 6 1 4 5 0 1 4 7 2 4 2 1 2 5 1 1 6 1 2 7 4 2 5 3 13 4 8 5 18 2 3 7 6 14 1 6 6 2 3 5 5 9 3 6 2 7 9 4 6 1 7 2 0 1 7 0

Output
Case 1: Path = 2 1 4; 8 second delay Case 2: Path = 1 2; 5 second delay Case 3: Path = 1 2 3 6 7; 20 second delay

Problem I. Intersecting line segments

Source file name:	I.c, I.cpp, I.java, I.py
Input:	`Standard`
Output:	`Standard`

In a 2-D Cartesian space, a straight line segment A is defined by two points $A_0 = (x_0, y_0)$, $A_1 = (x_1, y_1)$. The intersection of line segments A and B (if there is one), together with the initial four points, defines four new line segments. In Figure 1.1, the intersection P between lines B and C defines four new segments. As a result, the total amount of line segments after the evaluation of intersections is five.

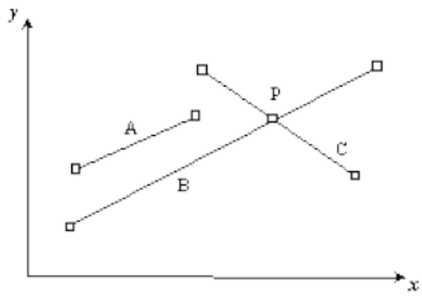


Figure 1.1 - Intersections of line segments

Given an initial set of lines segments, determine the number of line segments resulting from the evaluation of all the possible intersections. It is assumed, as a simplification, that no coincidences may occur between coordinates of singular points (intersections or end points).

Input

The input begins with a single positive integer on a line by itself indicating the number of test cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive test cases.

For each test case the first line contains the integer number N of line segments. Each of the following N lines contains four integer values x_0, y_0, x_1, y_1, separated by a single space, that define a line segment.

- $1 \leq N \leq 20000$
- $0 \leq x_0, y_0, x_1, y_1 \leq 10^6$

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line. The integer number of lines segments after all the possible intersections are evaluated.

Example

Input	Output
2 5 3 1 3 8 4 1 4 8 2 4 9 4 8 7 5 7 5 6 10 1 2 2 4 4 9 2 6 5 4	11 4

Problem J. Connected Components

Source file name:	J.c, J.cpp, J.java, J.py
Input:	`Standard`
Output:	`Standard`

A graph $G = (V, E)$ is connected if a path can be found in 0 or more intermediate steps between any pair of nodes in G. The following graph is not connected:

```
A ---- B        C ---- E
       |
       |
       D
```

It contains, however, many connected subgraphs, which are: $\{A\}$, $\{B\}$, $\{C\}$, $\{D\}$, $\{E\}$, $\{A, B\}$, $\{B, D\}$, $\{C, E\}$, $\{A, B, D\}$.

A connected subgraph is maximal if there are no nodes and edges in the original graph that could be added to the subgraph and still leave it connected. In the previous example, $\{C, E\}$ and $\{A, B, D\}$ are maximal.

Your job is to find, for the given graphs, how many connected subsets there are.

Input

There will be many input cases, each of which will be separated by a blank line. The last input case will be followed by EOF.

Each of the input cases starts with a line that contains a single upper case alphabetic character, which represents the largest node name in the graph. Each successive line contains a pair of upper case alphabetic characters denoting an edge in the graph. The graph represented by the input is undirected.

Output

For each test case print a single line with the number of maximal connected subgraphs.

Example

Input	Output
E AB CE DB B AB C	2 1 3

Problem K. Typo sequences

Source file name:	K.c, K.cpp, K.java, K.py
Input:	`Standard`
Output:	`Standard`

As part of a study about typos on modern smartphone keyboards, a volunteer is given a list of words and he is asked to type them all without correcting if they make a mistake. You've been hired to produce a program to analyze the results and determine which kinds of typos are more common.

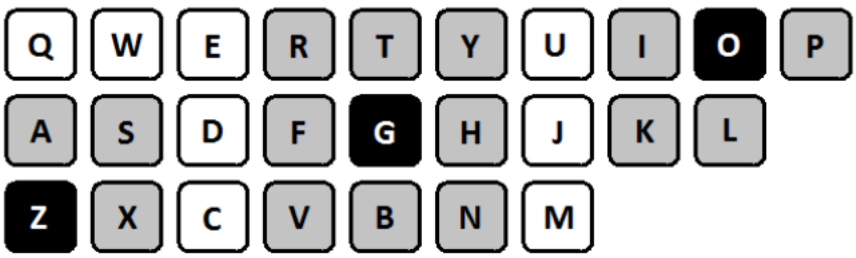
You're given a list of pairs of words, a reference word (the word the volunteer was supposed to type) and an output word (the word the volunteer actually typed).

For each pair of words, your job is to determine all the possible sequences of typos that the volunteer may have followed to produce the output word while attempting to type the reference word.

The volunteer attempted to type each letter of the reference word just once, and for each of those letters there are 5 possible outcomes when typing it:

1. OK (o): The volunteer typed the correct letter.
2. FORGOT (f): The volunteer forgot to type the letter.
   e.g. forgot to type an 'l' in 'hello', producing 'helo'.
3. WRONG (w): The volunteer typed an "adjacent" key instead of the correct one.
   e.g. typed 'r' instead of 'e' in 'hello' and produced 'hrllo'.
4. EXTRA ON LEFT (l): The volunteer pressed the correct key and an "adjacent" key together and the wrong letter was typed first.
   e.g. pressed keys for 'l' and 'i' when typing 'l' in 'hello', producing 'heillo'.
5. EXTRA ON RIGHT (r): The volunteer pressed the correct key and an "adjacent" key together and the correct letter was typed first.
   e.g. pressed keys for 'h' and 'y' when typing 'h' in 'hello', producing 'hyello'.

A key is "adjacent" to another if they share a border or a corner. See the image below.



Adjacent keys of 'z', 'g' and 'o'

As an example, if the reference word is 'normal' and the output word is 'nrtmsap', there is only one sequence of the above typos that could produce that output word ofrolw, because:

- 'n' is ok (o)
- 'o' is missing (f)

- 'r' followed by its adjacent 't' (r)
- 'm' is ok (o)
- 'a' preceded by its adjacent 's' (l)
- 'l' replaced by its adjacent 'p' (w)

There may be more than one sequence of typos that lead to the same output word.

As an example of such scenario if while typing 'allok', the volunteer produced 'allok' (yes, the same word, sequence ooooo), there can also be a sequence of typos that happen to produce the same word. ofowl, for example:

- 'a' is ok (o)
- 'l' is missing (f)
- 'l' is ok (o)
- 'o' replaced by its adjacent 'l' (w)
- 'k' preceded by its adjacent 'o' (l)

Input

The first line will only have a single positive integer K, smaller than 50. Then follow K pairs of words (each word in its own line), the first being the reference word and the second being the output word. Look at the sample.

The words are only formed by at most 15 lower case letters and there's always a sequence of typos that, when followed while attempting to type the reference word, can produce the output word.

While the reference word must have at least 1 character, the output word may be empty (since you can always forget to type all the letters).

Output

The output of each test case must be followed by a blank line. The output of a test case starts with a header on a single line, which consists of the reference word, followed by the output word, separated only by a single space, and finished with a colon (:). Look at the sample.

After the header, list on independent lines each possible sequence of typos that the volunteer could have made to type the output word while attempting to type the reference word.

Sequences must be ordered lexicographically.

Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>3
allok
allok
empty

normal
nrtmsap</pre></td><td><pre>allok allok:
ofolo
ofowl
ooflo
oofwl
ooofl
ooooo
ooorf
oorfo
oorwf

empty :
fffff

normal nrtmsap:
ofrolw</pre></td></tr>
</table>