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`

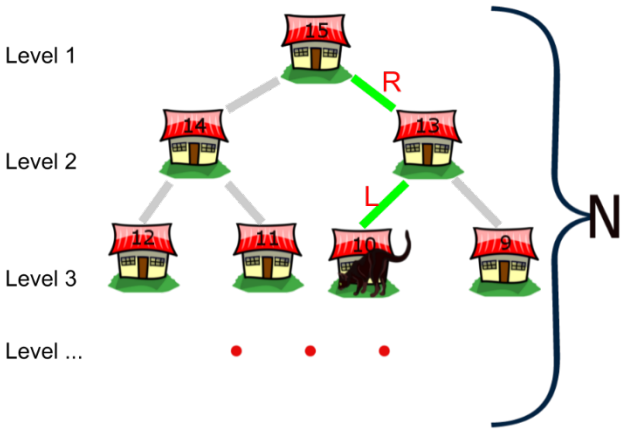
Baker is a very curious cat, this morning he went out of the neighborhood to explore the world, but, when he decided to come home he realized he was lost. Since baker is a very smart cat he remembers the number b of the house where we live.

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

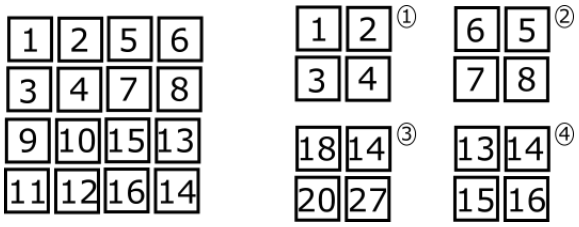
Input	Output
4 10 4 15 6 31	Instructions: RL Instructions: Instructions: LLLLR

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" $64 - 33 = 31$

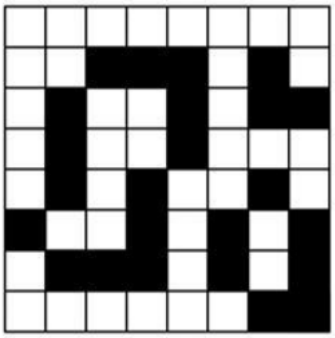


Figura 1. Ejemplo de un mapa de 8 x 8 celdas

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

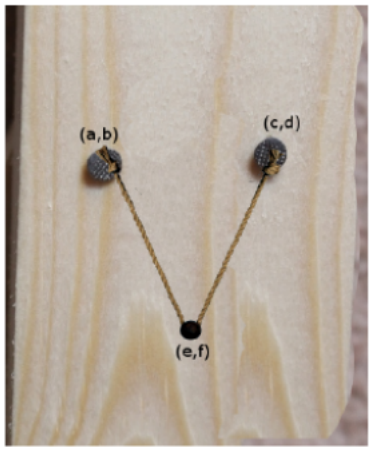
Source file name:	H.c, H.cpp, H.java, H.py
Input:	`Standard`
Output:	`Standard`

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 2 5 7 4 10 10 20 30 40 31	7 4 4.788675 0.169873 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.