The 2014 ACM-ICPC Asia Regional Contest
Xi'an Site

Sponsored by
IBM & Huawei & Huanlehuyu & iQIYI

Hosted by
Northwestern Polytechnical University
Xi'an, China

October 26, 2014

This problem set should contain eleven (11) problems on twenty (20) pages.
Please inform a runner immediately if something is missing from your problem set.

Problem A. Built with Qinghuai and Ari Factor

Description

DISCLAIMER: All names, incidents, characters and places appearing in this problem are fictitious. Any resemblance to actual events or locales or real persons, living or dead, is purely coincidental.

Shamatisan is a somewhat famous smartphone maker in China, they built smartphones that hope to contend with Abble and Dami for the hearts, minds and the wallets of China's consumers. They have a famous advertising word, saying that Shamatisan phones are built with Qinghuai (a concept which is hard to explain in English). Their latest phone T^{-1} has just began taking reservations recently, or to be precious, at the beginning of this month. But those who are tracking its progress on Aripapapa's online store Skyat noticed an interesting fact that has led to an apology by the online shopping site.

Those (being like sleuths in this story) who are always curious about questions like "In how many attoseconds[1] were the Dami phones sold out?" found something unusual about the reservation count of Shamatisan T^{-1}. It always has a divisor three! What's the logic behind this mystery? A bit of digging into the site coding showed that the number of reservations had been multipled by three. After this discovery, people started rumors like "Three is the Qinghuai factor, applied broadly by Shamatisan internally." and began to call integers, which are divisible by three, Qinghuai numbers. They also defined if all elements in a sequence are Qinghuai numbers, the sequence itself is said to be built with Qinghuai. Moreover, after some research, people found that there is a feature called "Buy Buy Buy Ring" on Skyat, causing all reservation counts multiplied by a factor (possibly 1). The rumor "Any real number can be represented as an Aripapapa Factor (also known as Ari Factor)" had been spread widely.

Later, an Aripapapa's spokeswoman said this was an incident and posted an official apology announcement. It is said that a programmer "made a highly unscientific decision". As a result, main programmer of Skyat whose name is Beiguoxia lost his job.

Our protagonist Pike loves to write programs that are able to automatically grab some data from Internet. As you may already know, such programs are usually called "spider".

Pike has already collected some sequence using his spider. Now he wonders that if these sequences are built with Qinghuai. Help Pike to figure out this!

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($1 \leq n \leq 100$), the length of sequence S. The second line contains n integers, which represent n integers in sequence S. All the numbers in the input will not exceed 10^6.

[1] 1 attosecond equals to 10^{-18} seconds.

Output

For each test case output one line "Case #x: y", where x is the case number (starting from 1) and y is "Yes" (without quotes) if the sequence S is built with so-called "Qinghuai", otherwise "No" (without quotes).

Samples

Sample Input	Sample Output
2 3 1 2 3 2 3000 996	Case #1: No Case #2: Yes

Hints

In the first case, since the sequence contains numbers which are too small to have Qinghuai, it cannot be called being built with Qinghuai.

In the second case, the first integer is the signage of Shamatisan, and the second integer represents core values of Aripapapa, we can declare that the sequence is built with Qinghuai.

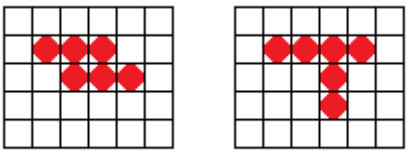
Also note that the whole problem statement (including hints) had deliberately been written as a joke, don't be so serious!

Problem B. Puzzle & Dragons

Description

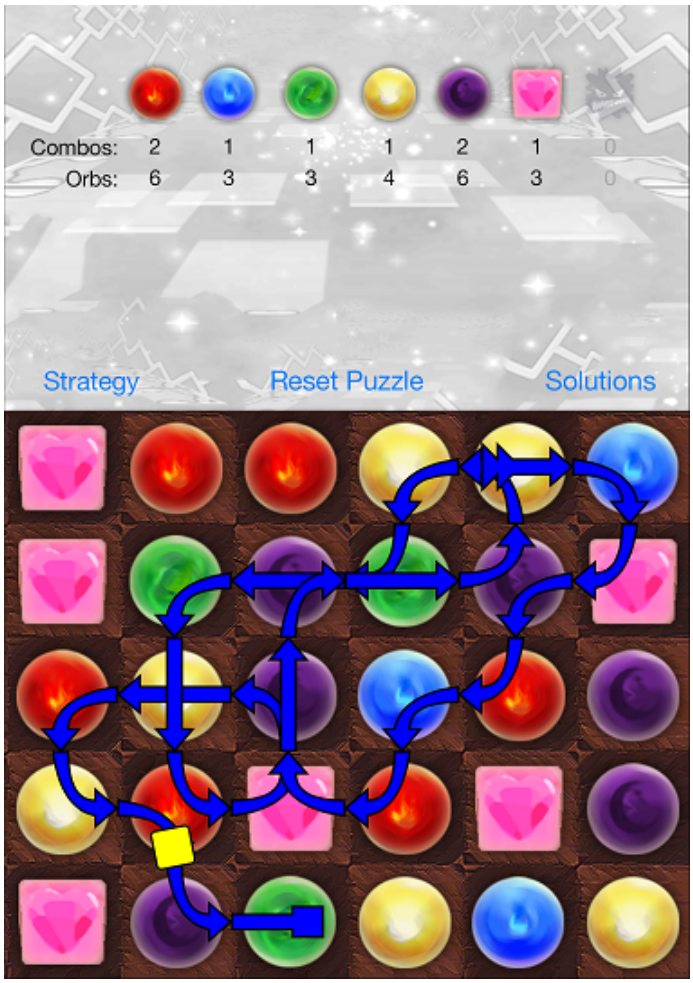
Recently, Little Apple is addicted to a game called "Puzzle & Dragons" (For short, PAD). There is a puzzle with 5×6 grid in this game, each square containing a drop. There are 6 types of drops: Fire, Water, Plant, Light, Darkness and Cure (represented by F,W,P,L,D,C respectively). You can see the picture below to get a more clear image of the game.

At the beginning of the game, the player can take one drop and move it through a path. The first drop on the path will move to the position of the second drop, the second drop will move to the position of the third drop, and so on. And the last drop will move to the position of the first drop. You can assume that the drops can only be moved up, down, left or right. For example, if a row of the puzzle is originally "FWPLDC", then the fire drop is taken and moved through a straight path to the right end, then this row will become "WPLDCF".



After the selection of the path, elimination begins. If there are 3 or more drops of same type connected in a row or a column (denoted by "a chain"), they will be eliminated, and the drops above them will fall down and no new drops will appear after elimination. This check and elimination will be repeated until there no longer exists chains.

If there are multiple chains eliminated after the movement, they will form "combos". It should be noticed that unlike some games with similar system, if two chains of same type are adjacent or share some common drops, they will be counted as just 1 combo. The figure below shows 2 examples of this case.



Now, it's your task to choose a path that can make most combos from all possible paths with length no longer than 9. If there is a tie, choose one that can eliminate most drops (as it will increase the damage and recovered HP). If a tie still exists, choose the path with shortest length. If still having multiple solutions, any of them would be OK.

Input

The first line of the input gives the number of test cases, T ($T \leq 100$). T test cases follow.

Each test case will contain 5 lines, each line contains 6 characters (either being F,W,P,L,D or C), representing the puzzle that you should solve.

Output

For each test case, first print a line containing "Case #x:", where x is the case number (starting from 1).

Then print three lines describing the solution:

The first line contains `Combo:`a `Length:`b, showing the number of combos you can achieve by this solution is a and the length of the path is b.

The second line contains two integers x and y, denoting that the drop in (x, y) will be taken (With the upper-left corner being $(1, 1)$ and the lower-right corner being $(5, 6)$).

The third line contains a string no longer than 9, each character being U, D, L or R, representing moving the drop up, down, left or right.

Samples

Sample Input	Sample Output
1 CFFLLW CPDPDC FLDWFD LFCFCD CDPLWL	Case #1: Combo:5 Length:9 4 3 RURURDLDD

Hints

After the move, the puzzle becomes:

$$\begin{matrix} C & F & F & L & L & W \\ C & P & D & P & C & D \\ F & L & D & F & C & D \\ L & F & F & W & W & D \\ C & D & P & L & C & L \end{matrix}$$

After first elimination(1 combo, 3 drops):

$$
\begin{array}{cccccc}
C & F & F & L & L & \\
C & P & D & P & C & \\
F & L & D & F & C & \\
L & F & F & W & W & W \\
C & D & P & L & C & L
\end{array}
$$

After second elimination(1 combo, 3 drops):

$$
\begin{array}{cccccc}
C & F & F & & & \\
C & P & D & L & L & \\
F & L & D & P & C & \\
L & F & F & F & C & \\
C & D & P & L & C & L
\end{array}
$$

After third elimination(2 combos, 6 drops):

$$
\begin{array}{cccccc}
C & & & & & \\
C & F & F & & & \\
F & P & D & L & & \\
L & L & D & P & & \\
C & D & P & L & L & L
\end{array}
$$

After forth elimination(1 combos, 3 drops):

$$
\begin{array}{cccc}
C & & & \\
C & F & F & \\
F & P & D & \\
L & L & D & L \\
C & D & P & P
\end{array}
$$

Problem C. The Problem Needs 3D Arrays

Description

A permutation is a sequence of integers $p_1, p_2, \ldots, p_n$, consisting of n distinct positive integers and each of them does not exceed n. Assume that $r(S)$ of sequence S denotes the number of inversions in sequence S (if $i < j$ and $S_i > S_j$, then the pair of (i, j) is called an inversion of S), $l(S)$ of sequence S denotes the length of sequence S. Given a permutation P of length n, it's your task to find a subsequence S of P with maximum $\frac{r(S)}{l(S)}$. A subsequence of P is a sequence $(p_{i_1}, p_{i_2}, \ldots, p_{i_t})$ which satisfies that $0 < i_1 < i_2 < \ldots < i_t \leq n$.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($1 \leq n \leq 100$), the length of the permutation P. The second line contains n integers $p_1, p_2, \ldots, p_n$, which represents the permutation P.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the maximum $\frac{r(S)}{l(S)}$.

Your answer will be considered correct if it is within an absolute error of 10^{-6} of the correct answer.

Samples

Sample Input	Sample Output
1 5 3 4 2 5 1	Case #1: 1.250000000000

Problem D. The Diameter of Tree

Description

One day, Little Apple drew a tree on a paper and he wrote down the DFS (depth-first search) sequence and the BFS (breadth-first search) sequence of the tree. After a few days, he wants to know the diameter of the tree he drew. Sadly, the paper with the tree is lost. He can only remember the DFS sequence and the BFS sequence of the tree so that he wants to know the expected diameter length of the tree. However, he has no idea how to get it. As an excellent programmer, you are asked for help.

Assume that S is the vertex set of the tree. The distance between two vertices u, v is the length (in edges) of the shortest path between vertice u and vertice v. The diameter of a tree is equal to

$$\max\{\text{dist}(u, v) \mid u, v \in S\}$$

Here $\text{dist}(u, v)$ denotes the distance between two vertices u and v.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($1 \leq n \leq 10000$), the number of vertices of the tree. Then two lines follow. The first line contains n integers, which represent the DFS sequence of the tree. The second line also contains n integers, which represent the BFS sequence of the tree.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the expected diameter of the tree.

Your answer will be considered correct if it is within an absolute error of 10^{-4} of the correct answer.

Samples

Sample Input	Sample Output
1 7 1 2 3 5 4 7 6 1 2 4 6 3 5 7	Case #1: 4.000

Problem E. Brushing King

Description

Mr. Big is one of students of Brushing King. He is always sleepy when the course begins. But Brushing King will punish the students who sleep in the class. In order not to be brushed by Brushing King, Big wants to know if there is a safe place for him to sleep through the whole class.

The Brushing King could be considered as a point. His sight is considered as a circular sector with angle θ and radius R.

Big selected several positions to sleep and he wants to know which one will not be seen by Brushing King during the course.

Brushing King always walks towards a direction at a speed of 1 and the direction vector of his movement will be given. He may rotate his sight or his speed direction at some moment. The course ends right after his last action.

Note that once the position is in the sight of Brushing King (even when Brushing King just rotates his head), Mr. Big will be caught by Brushing King. Hitting on the edge of the sight will be also considered as being seen.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains n, m, θ, R ($1 \le n, m, R \le 1000$, $0 < \theta < 180$) means the number of position Mr. Big selected and the number of Brushing King's actions. The angle will be given in degree.

The second line contains six integers p_x, p_y, v_x, v_y, d_x, d_y.

(p_x, p_y) is the initial position of Brushing King, (v_x, v_y) indicates the initial direction vector from the initial position of Brushing King to the midpoint of the arc of his sight sector (Note that it's **NOT** guaranteed $|(v_x, v_y)|, |(d_x, d_y)|$ equals to 1 or R.). (d_x, d_y) is the **direction vector** of Brushing King's movement. ($-2000 \le p_x, p_y \le 2000$, $1 \le |(v_x, v_y)|, |(d_x, d_y)| \le 2000$).

Then n lines follow, each of which contains two integers x, y ($-2000 \le x, y \le 2000$) means the coordinate of position selected by Mr.Big.

Then m lines follow, each of which contains three integers, p, t, α ($0 \le t \le 2000$, $0 \le \alpha \le 180$). There are two types of actions:

1. $p = 1$ means that at time t, Brushing King rotates the direction vector of his sight (i.e. the direction vector from his position to the midpoint of the arc of the sight sector) by α degrees clockwise.
2. $p = 2$ means that at time t, Brushing King rotates the direction vector of his movement by α degrees clockwise.

All the t will be given in strictly increasing order. Brushing King's actions are incredibly fast and they could be treated as finished in no time.

Output

For each test case output one line containing "Case #x:", where x is the test case number (starting from 1), followed by n numbers, each number is 0 or 1. The i-th number is 1 means that Mr. Big can survive in i-th position, otherwise 0.

Samples

Sample Input	Sample Output
1 3 2 90 3 0 0 0 1 0 1 50 1 -1 0 -100 0 1 1 180 1 100 0	Case #1: 101

Problem F. Color

Description

Recently, Mr. Big recieved n flowers from his fans. He wants to recolor those flowers with m colors. The flowers are put in a line. It is not allowed to color any adjacent flowers with the same color. Flowers i and $i+1$ are said to be adjacent for every i, $1 \leq i < n$. Mr. Big also wants the total number of different colors of the n flowers being exactly k.

Two ways are considered different if and only if there is at least one flower being colored with different colors.

Input

The first line of the input gives the number of test cases, T. T test cases follow. T is about 300 and in most cases k is relatively small.

For each test case, there will be one line, which contains three integers n, m, k ($1 \leq n, m \leq 10^9$, $1 \leq k \leq 10^6$, $k \leq n, m$).

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the number of ways of different coloring methods modulo $10^9 + 7$.

Samples

Sample Input	Sample Output
2 3 2 2 3 2 1	Case #1: 2 Case #2: 0

Problem G. The Problem to Slow Down You

Description

After finishing his homework, our problem setter Federmann decided to kill time by hanging around online. He found a cool chat room that discusses competitive programming. Federmann has already joined lot of such chat rooms, but this one is special. Once he entered the chat room, he noticed that there is an announcement saying "We forbid off-topic messages!". Federmann thinks that's quite unusual, he decided to sit down and join the talk. After watching people discussing different programming challenges for a while, he found an interesting message saying "No, Federmann won't prepare another problem about strings this year."

"Oh, why do you guys think about that?" Federmann smiled. "Don't they know I have an Edward number[2] of 3?"

He then thought about something about palindrome, given two strings A and B, what is the number of their common palindrome substrings? The amount of common palindrome substrings between two strings is defined as the number of quadruple (p, q, s, t), which satisfies that:

1. $1 \le p, q \le length(A)$, $1 \le s, t \le length(B)$, $p \le q$ and $s \le t$. Here $length(A)$ means the length of string A.
2. $A_{p..q} = B_{s..t}$
3. $A_{p..q}$ is palindrome. (palindrome string is the string that reads the same forward or backward)

For example, $(1, 3, 1, 3)$ and $(1, 3, 3, 5)$ are both considered as a valid common palindrome substring between `aba` and `ababa`.

Federmann is excited about his new task, and he is just too lazy to write solutions, help him.

Input

The first line of the input gives the number of test cases, T. T test cases follow. For each test case, the first line contains a string A and the second line contains a string B. The length of A, B will not exceed 200000.

It is guaranteed the input file will be smaller than 8 MB.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the number of common palindrome substrings of A and B.

[2]The Edward number is something like Erdős number, among problem setters.

Samples

Sample Input	Sample Output
3 abacab abccab faultydogeuniversity hasnopalindromeatall abbacabbaccab youmayexpectedstrongsamplesbutnow	Case #1: 12 Case #2: 20 Case #3: 18

Problem H. The Problem to Make You Happy

Description

Alice and Bob are good friends, as in every other storyline. One day Alice and Bob are playing an interesting game. The game is played on a **directed** graph with n vertices and m edges, Alice and Bob have exactly one chess piece each. Initially, Bob's chess piece is placed on vertex x, while Alice's chess piece is placed at vertex y. Bob plays first, then Alice, then Bob, then Alice and so on.

During each move, the player must move his/her chess piece from the vertex his/her chess piece currently at to an adjacent vertex, by traveling through exactly one directed edge. (Remember that the game is played on a **directed** graph.) If someone can't make such a move, he/she will be considered to lose the game.

There's one additional rule: at any time, if Bob and Alice's chess pieces are at the same vertex, then Alice is consider to be the winner and the game ends up immediately.

Now you are given the initial situation, could you determine who is the winner? Please note that neither Bob nor Alice will make any mistakes during the game, i.e. they both play optimally. In case that the game never ends up, Bob is considered to be the winner.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains two integers n and m ($2 \leq n \leq 100$, $1 \leq m \leq n \times (n-1)$). Next m lines, each line contains two integers b and e, indicating there is one directed edge from vertex b to vertex e. Last line contains two integers x and y ($1 \leq x, y \leq n, x \neq y$), which are Bob and Alice's initial position. The graph contains no self-loops or duplicate edges.

Output

For each test case output one line "Case #x: y", where x is the case number (starting from 1) and y is "Yes" (without quotes) if Bob can win or the game never ends up, otherwise "No" (without quotes).

Samples

Sample Input	Sample Output
3 5 3 1 2 3 4 4 5 3 1 4 3 1 2 2 3 3 4 1 2 3 3 1 2 2 3 3 1 2 1	Case #1: Yes Case #2: No Case #3: Yes

Problem I. International Collegiate Routing Contest

Description

You may know that Bluegao University (formly Bluefly University) is famous of networking technology. One day, their headmaster Yuege received a special router, along with a task about routing table.

In this problem, routing table is a (probably) big table with several items, each item represents a subnet. The router has limited function, it can only deal with two next-hops and one main routing table. Packets will be send to next hop A if there exists a subnet containing the destination of the packet in the main routing table. Otherwise they will be send to next hop B.

You may know that, IPv4 uses 32-bit (four-byte) addresses, which limits the address space to 4294967296 (2^{32}) addresses. IPv4 addresses may be written in any notation expressing a 32-bit integer value, for human convenience, they are most often written in the dot-decimal notation, which consists of four octets of the address expressed individually in decimal and separated by periods. But their binary notation is also very useful. For example, IP address 128.2.142.23 can be expressed in dot-binary notation as 10000000.00000010.10001110.00010111. A subnet is a block of adjacent IP addresses with exactly same binary prefix, and usually written as the first IP address in its address space together with the bit length of prefix, like "202.120.224.0/24". If an IP address is in the range of an subnet, we say that this subnet contains the IP address.

Yuege's task is to invert the behaviour of his router, make all packets currently routed to hop A route to hop B, and vice versa. Also he wants to keep the size of the main routing table as small as possible, for performance.

In short, for a given routing table (i.e. a bunch of subnets), we need to get its "complement", i.e. calculate a **minimum** set of subnets which have no intersection with given subnets, and their union must be the whole IPv4 address space.

Remember that Bluegao University is famous of networking tech, as headmaster of Bluegao University, Yuege definitely knows how to solve such problem, but he is too lazy to code, so he turns to you for help.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($0 \leq n \leq 30000$), the number of subnets. Next n lines, each contains a subnet in the format of $a.b.c.d/l$, a, b, c, d, l are all integers, $0 \leq a, b, c, d < 256$, $0 \leq l \leq 32$.

Note that even if $l = 32$, the "/32" part should not be omitted. And if $l = 0$, the IP address part must be "0.0.0.0".

Output

For each test case, first output one line "Case #x:", where x is the case number (starting from 1). Then on the second line print an integer n indicates the number of subnets in your answer. Next n lines each line contains a subnet in the same format as input. You may output them in any order.

Samples

Sample Input	Sample Output
3 0 1 0.0.0.0/1 1 128.0.0.0/1	Case #1: 1 0.0.0.0/0 Case #2: 1 128.0.0.0/1 Case #3: 1 0.0.0.0/1

Problem J. Unlimited Battery Works

Description

One day, Little Apple invented a chess game on a rooted tree and he wants to play with you. At the beginning, there is one and only one black chess piece placed on every vertex of the tree. In each turn, you can cast a magic on one vertex, which will change the chess piece on it into white piece whatever it's white or black before magic. And at the same time, chess pieces on some vertices will also be changed into white pieces. If you cast a magic on vertex i, chess piece on vertex j will also be changed into white piece if and only if vertex j is in the subtree of vertex i and the length (in edges) of the shortest path between i and j is no more than A_i.

Little Apple wants to know how many steps are neccessary in order to change the tree into a tree with all chess pieces white. However, as an excellent programmer, you think it's too easy for you. You want to calculate the expected steps needed to change all chess pieces on the tree into white if you choose a vertex to cast a magic randomly (assume that each vertex has the same probability to be chosen).

Please notice that:

1. If all chess pieces on the tree have been changed into white, you won't cast magic anymore.
2. In every turn, it's possible to choose **any** vertex of the tree.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($1 \leq n \leq 50$), the number of vertices on the tree. The second line contains n integers, which represent $A_1, A_2, \ldots, A_n$ (as described above, $1 \leq A_i < 50$ for $1 \leq i \leq n$). The last line conatins $n - 1$ integers, the i-th integer represent the parent of vertex $i + 1$.

It guaranteed that the given graph is a tree and vertex 1 is always the root of the tree.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the expected steps needed to change all the chess pieces into white.

Your answer will be considered correct if it is within an absolute error of 10^{-6} of the correct answer.

Samples

Sample Input	Sample Output
3 6 1 0 0 0 0 0 1 1 1 1 1 6 1 0 1 0 1 0 1 2 3 4 5 50 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 4 3 1 2 7 6 9 6 8 10 9 13 16 15 13 18 14 15 19 22 18 24 26 27 25 27 28 25 28 30 34 34 33 34 34 33 36 36 36 37 42 42 44 43 46 48	Case #1: 6.000000000000 Case #2: 11.000000000000 Case #3: 224.960266916471

Problem K. Last Defence

Description

Given two integers A and B. Sequence S is defined as follow:

- $S_0 = A$
- $S_1 = B$
- $S_i = |S_{i-1} - S_{i-2}|$ for $i \geq 2$

Count the number of distinct numbers in S.

Input

The first line of the input gives the number of test cases, T. T test cases follow. T is about 100000.

Each test case consists of one line - two space-separated integers A, B. $(0 \leq A, B \leq 10^{18})$.

Output

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and y is the number of distinct numbers in S.

Samples

Sample Input	Sample Output
2 7 4 3 5	Case #1: 6 Case #2: 5