Problem A: Abstract Names

Some of you may have noticed that in certain computer games, particularly the ones based on sports, the spelling of names are mutated so that they are not an exact duplicate of the real entity. This is done to avoid hassles of taking permission from each player as well as any patent issues. In this problem, you will be given a pair of names, one of which is that of a player in real life and the second found in a game. You will have to determine if the two names are same, that is the second one is obtained by mutating the first.

Two names are considered same if they are of same length and they only vary at positions where vowels occur. That means, a name which can be obtained by replacing zero or more vowels by other vowels to obtain a new name are considered same, provided they have same length. For example, both *polo* and *pola* are same as *pele* but not *pelet* or *bele*.

Input

First line of input contains a positive integer **n ≤ 20,** where **n** denotes the number of test cases. This will be followed by **2*n** lines where each line will contain a name of at most 20 characters. The names will consist of lower case letters only.

Output

For each case of input, there will be one line of output. It will be *Yes* if the second name can be obtained by mutating the first name, otherwise it will be *No*.

Sample Input	Output for Sample Input
5 pele polo pele pola ronaldo ronaldino pele pelet pele bele	Yes Yes No No No

Problem Setter: Shamim Hafiz
Special Thanks: Tanveer Ahsan

Problem B: Blind Sorting

I am a polar bear. But I am not just an ordinary polar bear. Yes I am extra ordinary! I love to play with numbers. One day my very good friend Mr. Panda came to me, and challenged me to solve a puzzle. He blindfolded me, and said that I have **n** distinct numbers. What I can ask is whether a'th number is larger than b'th number and he will answer me properly. What I have to do is to find out the largest and second largest number. I thought for a while and said "Come on, I will do it in minimum number of comparison."

Input

There will be a non-negative integer, **n** in each of the line of input where **n** is as described above. **n** will be less than any 10 digit prime number and not less than the smallest prime.

Output

For each **n**, output number of questions that I have to ask Mr. Panda in the worst case.

Sample Input	Output for Sample Input
2 4	1 4

Problem Setter: Mahbubul Hasan
Special Thanks: Shamim Hafiz

Problem C: Car

You are in a car and going at the speed of **u** m/s. Your acceleration **a** is constant. After a particular time **t,** your speed is **v** m/s and your displacement is **s**. Now you are given some (not all of them) values for the given variables. And you have to find the missing parameters.

Input

The input file may contain multiple test cases. Each test case can be one of the

```
1 u v t
2 u v a
3 u a s
4 v a s
```

Input will be terminated by a single **0**.

Output

For each case of input you have to print one line containing the case number and

If `1 u v t` are given then print **s** and **a**
If `2 u v a` are given then print **s** and **t**
If `3 u a s` are given then print **v** and **t**
If `4 v a s` are given then print **u** and **t**

Check the samples for more details. You can assume that the given cases will not evaluate to an invalid situation. Use 'double' for all calculations and output all floating point numbers to three decimal places.

Sample Input	Output for Sample Input
1 10 5 2.0	Case 1: 15.000 -2.500
1 5 10.0 2	Case 2: 15.000 2.500
2 10 11 2	Case 3: 5.250 0.500
3 5 1 6	Case 4: 6.083 1.083
4 5.0 -1 6	Case 5: 6.083 1.083
0	

Problem Setter: Jane Alam Jan
Special Thanks: Md. Shiplu Hawlader

Problem D: Digital Fortress

In the last IIUPC there was a problem called Da Vinci Code prepared on the story of the bestselling book of Dan Brown, *The Da Vinci Code*. Here is another problem based on his another techno-thriller novel *Digital Fortress*. In this problem, you will be given a cipher text. Your task is to decipher the text using the decrypting technique described below. Let's take an example. A cipher text is given as follows:

WECGEWHYAAIORTNU

The output will be:

WEAREWATCHINGYOU

For this problem, there are **16** characters in the given cipher text "**WECGEWHYAAIORTNU**" which is square of **4**. These letters have to be arranged in **n × n** (in this example **4 × 4**) grid and each letter from the given input will be placed in a grid in row major order (1st row, 2nd row, 3rd row, …). When the given cipher text is placed in the grid it looks like as follow:

```
W E C G
E W H Y
A A I O
R T N U
```

From the above grid if we take the letters in column major order (1st column, 2nd column, 3rd column, …) then we get the following decrypted text:

WEAREWATCHINGYOU

Input

Input starts with a line consisting of a single number **T. T** test cases follow. Each test case consists of one line. This line contains the cipher text. The cipher text contains either UPPERCASE letters or blank spaces. Total number of character in the text will not be more **10,000**.

Output

For each test case, the output contains a single line containing the decrypted text. If the number of characters in the input text is not square of any number, then give the output "INVALID".

Sample Input	Output for Sample Input
3 WECGEWHYAAIORTNU DAVINCICODE DTFRIAOEGLRSI TS	WEAREWATCHINGYOU INVALID DIGITAL FORTRESS

Problem Setter: Mohammed Shamsul Alam
Special Thanks: Tanveer Ahsan

Problem E: Energy Saving Microcontroller

The rapid progress of the electronics era is mostly due to wide availability of microcontrollers. They are used in many devices, particularly in stand alone devices that need to do real time processing. Some common uses are in cell phones, hand held video games and pace makers. A lot of these stand alone devices run on internal battery power with no connection to external power source when in operation. Therefore it is critical that, the devices are able to minimize their power usage when idle. In this problem, you will be required to simulate the behavior of a simply designed microcontroller based device.

Input

The first line of input contains a positive integer **T** **(≤10)**, where **T** denotes the total number of scenarios to simulate. Each scenario starts with three positive numbers **n** **(≤1000)**, **i** and **k**. Both **i** and **k** will fit in **32 bit signed integer**. The next **n** lines contain a positive integer each. These numbers will be in increasing order and will fit in 32 bit signed integer. These numbers indicate the time in microseconds when an instruction is sent to the microcontroller for execution. If the microcontroller is in active state, it will execute the instruction instantly. In the event the microcontroller is inactive, it will take **k** microseconds to activate itself and then it will process the instruction. If any instructions are sent during the activation period, those will be ignored. The microcontroller is active at the 0^{th} microsecond, that is after it starts its operation. After that, if it doesn't execute any instruction for at least **i** microseconds, it goes into inactive state. Note that, the microcontroller may go into inactive state anytime it doesn't receive an instruction for at least **i** microseconds, after processing an instruction. If an instruction comes just when the device is about to go idle, the device will go idle and then the instruction will activate the device.

Output

For each input scenario, there will be one line of output. It will contain the case number followed by the number of times the microcontroller went into inactive state and the number of instructions that were ignored; the two numbers will be separated by a single space. Please look at sample output for exact formatting.

Sample Input	Output for Sample Input
3 1 2 3 1 2 2 3 10 11 2 2 1 10 11	Case 1: 0 0 Case 2: 1 1 Case 3: 1 0

Problem Setter: Shamim Hafiz
Special Thanks: Sohel Hafiz

Problem F: Fantasy of a Summation

If you think codes, eat codes then sometimes you may get stressed. In your dreams you may see huge codes, as I have seen once. Here is the code I saw in my dream.

```
#include <stdio.h>

int cases, caseno;
int n, K, MOD;
int A[1001];

int main() {
  int i, i1, i2, i3, ... , iK;

  scanf("%d", &cases);
  while( cases-- ) {
    scanf("%d %d %d", &n, &K, &MOD);
    for( i = 0; i < n; i++ ) scanf("%d", &A[i]);

    int res = 0;
    for( i1 = 0; i1 < n; i1++ ) {
      for( i2 = 0; i2 < n; i2++ ) {
        for( i3 = 0; i3 < n; i3++ ) {
          ...
            for( iK = 0; iK < n; iK++ ) {
              res = ( res + A[i1] + A[i2] + A[i3] + ... + A[iK] ) % MOD;
            }
          ...
        }
      }
    }
    printf("Case %d: %d\n", ++caseno, res);
  }
  return 0;
}
```

Actually the code was about - 'You are given 3 integers **n**, **K**, **MOD** and **n** integers – A_0, A_1, A_2, ... , A_{n-1}. You have to write **K** nested loops and calculate the summation of all A_i where **i** is the value of any nested loop variable.'

Now you have to find the result according to the code.

Input

The first line of input contains **T** denoting the number of cases.

Each case starts with three integers – **n** ($1 \le n \le 1000$), **K** ($1 \le K < 2^{31}$), **MOD** ($1 \le MOD \le 35000$). The next line will contain **n** non-negative integers denoting A_0, A_1, A_2, ... , A_{n-1}. Each of these integers will be fit into a 32 bit signed integer.

Output

For each case print the case number and the result. Follow the sample output for the exact output format.

Sample Input	Output for Sample Input
2 3 1 35000 1 2 3 2 3 35000 1 2	Case 1: 6 Case 2: 36

Problem Setter: Jane Alam Jan
Special Thanks: Anna Fariha

Problem G: Gridland Airports

The country 'Gridland' is a strange country which have **R*C** cities arranged in **R** rows and **C** columns. The bottom-left city is **(1,1)** and the upper-right city is **(R,C)**.

The governor of 'Gridland' has hired you to assign some flight routes between the cities in 'Gridland'. You have to establish minimum number of direct flight connections such that every pair of city is connected by a direct flight or some flight sequences. If a flight connection is established between two cities that can be used in both directions. But there is a restriction: a direct flight connection between two cities **A(r1, c1)** and **B(r2, c2)** can be established only if **|r1-r2|+|c1-c2| == odd**.

Find how many ways you can setup flight routes such that every pair of city is connected and the number of direct flight connections is minimum.

Input

First line of input is **T(≤5000)** which is the number of cases. Then there are **T** lines each containing two numbers **R**, **C** and **2 ≤ R, C ≤ 10^8**.

Output

Output the number of ways to setup the flight route network. As the answer could be very big so output answer **MOD (10^16+7)**.

Sample Input	Output for Sample Input
2 2 2 49 49	4 1661809100947531

Problem Setter: Tasnim Imran Sunny
Special Thanks: Md. Mahbubul Hasan, Md. Arifuzzaman Arif

Problem H: How Many Ways

Dexter has **N** coins having values **1,2,3, … N**. He should select a subset of exactly **K** coins from those such that the selected coins sum to **N**. Find how many ways he can do it. Suppose, N=8, K=3 then he can select coins in 2 ways: {1,2,5}, {1,3,4}.

Input

First line of input is **T** **(≤20)** which is the number of cases. Then there are **T** lines each containing two numbers **K (1 ≤ K ≤ 10)** and **N (1 ≤ N ≤ 10^9)**.

Output

Output the number of ways to choose **K** coins **MOD 1000000007**.

Sample Input	Output for Sample Input
3 4 10 3 8 4 231	1 2 80142

Problem Setter: Tasnim Imran Sunny
Special Thanks: Mahbubul Hasan

Problem I: Instant View of Big Bang

Have you forgot about wormholes? Oh my god! Ok, let me explain again.

A wormhole is a subspace tunnel through space and time connecting two star systems. Wormholes have a few peculiar properties:

1. Wormholes are one-way only.
2. The time it takes to travel through a wormhole is negligible.
3. A wormhole has two end points, each situated in a star system.
4. A star system may have more than one wormhole end point within its boundaries.
5. Between any pair of star systems, there is at most one wormhole in each direction.
6. There are no wormholes with both end points in the same star system.

All wormholes have a constant time difference between their end points. For example, a specific wormhole may cause the person traveling through it to end up 15 years in the future. Another wormhole may cause the person to end up 42 years in the past.

A brilliant physicist, wants to use wormholes to study the Big Bang. Since warp drive has not been invented yet, it is not possible for her to travel from one star system to another one directly. This *can* be done using wormholes, of course.

The scientist can start her journey from any star system. Then she wants to reach a cycle of wormholes somewhere in the universe that causes her to end up in the past. By traveling along this cycle a lot of times, the scientist is able to go back as far in time as necessary to reach the beginning of the universe and see the Big Bang with her own eyes. Write a program to help her to find such star systems where she can start her journey.

Input

The first line of the input will contain **`T,`** denoting the number of cases.

Each case starts with a blank line. The next line contains two numbers **`n`** and **`m`** . These indicate the number of star systems $(1 \le n \le 1000)$ and the number of wormholes $(0 \le m \le 2000)$. The star systems are numbered from **`0`** through **`n-1`** . For each wormhole a line containing three integer numbers **`x`**, **`y`** and **`t`** is given. These numbers indicate that this wormhole allows someone to travel from the star system numbered **`x`** to the star system numbered **`y`**, thereby ending up **`t`** $(-1000 \le t \le 1000)$ years in the future or past, a negative integer denotes past, positive integer denotes future.

Output

For each case, print the case number first, then print the star systems (in ascending order) where she can start her journey. If no such star system is found, print **`'impossible'`**.

Sample Input	Output for Sample Input

`2`
 	
`3 3`
`0 1 1000`
`1 2 15`
`2 1 -42`
 	
`4 4`
`0 1 10`
`1 2 20`
`2 3 30`
`3 0 -60`
	`Case 1: 0 1 2`
	`Case 2: impossible`

Problem Setter: Jane Alam Jan
Special Thanks: Md. Abirul Islam

Problem J: Joining with Friend

You are going from Dhaka to Chittagong by train and you came to know one of your old friends is going from city Chittagong to Sylhet. You also know that both the trains will have a stoppage at junction Akhaura at almost same time. You wanted to see your friend there. But the system of the country is not that good. The times of reaching to Akhaura for both trains are not fixed. In fact your train can reach in any time within the interval **[t1, t2]** with equal probability. The other one will reach in any time within the interval **[s1, s2]** with equal probability. Each of the trains will stop for w minutes after reaching the junction. You can only see your friend, if in some time both of the trains is present in the station. Find the probability that you can see your friend.

Input

The first line of input will denote the number of cases **T** (**T < 500**). Each of the following **T** line will contain **5** integers **t1, t2, s1, s2, w (360 ≤ t1 < t2 < 1080, 360 ≤ s1 < s2 < 1080 and 1 ≤ w ≤ 90)**. All inputs **t1**, **t2**, **s1**, **s2** and **w** are given in minutes and **t1, t2, s1, s2** are minutes since midnight **00:00**.

Output

For each test case print one line of output in the format "**Case #k: p**" Here **k** is the case number and **p** is the probability of seeing your friend. Up to **1e-6** error in your output will be acceptable.

Sample Input	Output for Sample Input
2 1000 1040 1000 1040 20 720 750 730 760 16	Case #1: 0.75000000 Case #2: 0.67111111

Problem Setter: Md. Towhidul Islam Talukder
Special Thanks: Samee Zahur, Mahbubul Hasan