WEEK – 3

Problem - 1

Count on Cantor

Problem code: CANTON

One of the famous proofs of modern mathematics is Georg Cantor's demonstration that the set of rational numbers is enumerable. The proof works by using an explicit enumeration of rational numbers as shown in the diagram below.

```
1/1 1/2 1/3 1/4 1/5 ...
2/1 2/2 2/3 2/4
3/1 3/2 3/3
4/1 4/2
5/1
```

In the above diagram, the first term is 1/1, the second term is 1/2, the third term is 2/1, the fourth term is 3/1, the fifth term is 2/2, and so on.

Input

The input starts with a line containing a single integer **t** <= 20, the number of test cases. **t** test cases follow.

Then, it contains a single number per line.

Output

You are to write a program that will read a list of numbers in the range from 1 to 10^7 and will print for each number the corresponding term in Cantor's enumeration as given below.

Example

```
Input:
3
3
14
7

Output:
TERM 3 IS 2/1
TERM 14 IS 2/4
TERM 7 IS 1/4
```

Problem - 2

Ambiguous Permutations

Problem code: PERMUT2

Some programming contest problems are really tricky: not only do they require a different output format from what you might have expected, but also the sample output does not show the difference. For an example, let us look at permutations.

A **permutation** of the integers *1* to *n* is an ordering of these integers. So the natural way to represent a permutation is to list the integers in this order. With *n = 5*, a permutation might look like 2, 3, 4, 5, 1.

However, there is another possibility of representing a permutation: You create a list of numbers where the *i*-th number is the position of the integer *i* in the permutation. Let us call this second possibility an **inverse permutation**. The inverse permutation for the sequence above is 5, 1, 2, 3, 4.

An **ambiguous permutation** is a permutation which cannot be distinguished from its inverse permutation. The permutation 1, 4, 3, 2 for example is ambiguous, because its inverse permutation is the same. To get rid of such annoying sample test cases, you have to write a program which detects if a given permutation is ambiguous or not.

Input Specification

The input contains several test cases.

The first line of each test case contains an integer *n* (*1 ≤ n ≤ 100000*). Then a permutation of the integers *1*to *n* follows in the next line. There is exactly one space character between consecutive integers. You can assume that every integer between *1* and *n* appears exactly once in the permutation.

The last test case is followed by a zero.

Output Specification

For each test case output whether the permutation is ambiguous or not. Adhere to the format shown in the sample output.

Sample Input

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

Sample Output

```
ambiguous
not ambiguous
ambiguous
```

Problem - 3

Alphacode

Problem code: ACODE

Nestor was doing the work of his math class about three days but he is tired of make operations a lot and he should deliver his task tomorrow. His math's teacher gives two numbers a and b. The problem consist in find the last digit of the potency of base a and index b. Help Nestor with his problem. You are given two integer numbers: the base a (0 <= a <= 20) and the index b (0 <= b <= 2,147,483,000), a and b both are not 0. You have to find the last digit of a^b.

Input

The first line of input contains an integer t, the number of test cases (t <= 30). t test cases follow. For each test case will appear a and b separated by space.

Output

For each test case output an integer per line representing the result.

Example

Input:

```
2
3 10
6 2
```

Output:

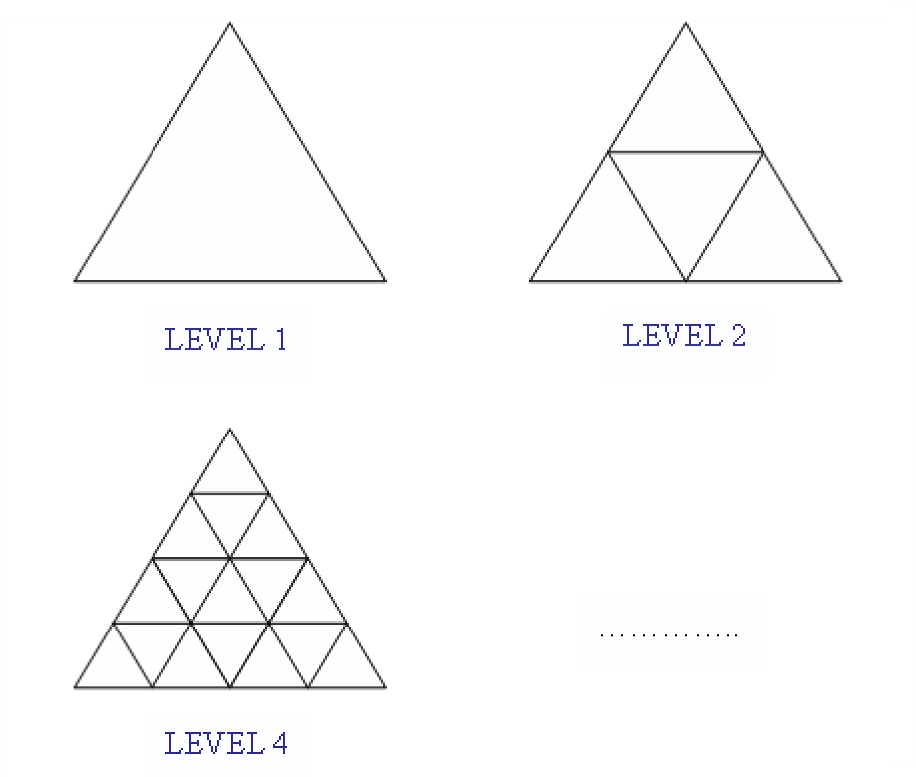
```
9
6
```

Problem - 4

Counting Triangles

Problem code: TRICOUNT

We define the LEVEL of a triangle as in the following illustrative image:



Task: Your task is very easy. All you have to do is to count all triangles in the biggest one (Level N).

Input

The first line of the input contains an integer T (T ≤ 10000) - the number of test cases and T lines follow. Each line contains an integer N ($1 \le N \le 10^6$) which is the level of the triangle in that test case.

Output

For each test case, you should write a seperate line: the number of triangles in the biggest one (Level N). (All answers will fit within the range of a 64-bit integer)

Example

Input:

```
3
1
2
3
```

Output:

```
1
5
13
```

Problem - 5

Stamps

Problem code: STAMPS

Everybody hates Raymond. He’s the largest stamp collector on planet earth and because of that he always makes fun of all the others at the stamp collector parties. Fortunately everybody loves Lucy, and she has a plan. She secretly asks her friends whether they could lend her some stamps, so that she can embarrass Raymond by showing an even larger collection than his. Raymond is so sure about his superiority that he always tells how many stamps he’ll show.And since Lucy knows how many she owns, she knows how many more she needs. She also knows how many friends would lend her some stamps and how many each would lend. But she’s like to borrow from as few friends as possible and if she needs too many then she’d rather not do it at all. Can you tell her the minimum number of friends she needs to borrow from?

Input

The first line contains the number of scenarios. Each scenario describes one collectors party and its first line tells you how many stamps (from 1 to 1000000) Lucy needs to borrow and how many friends (from 1 to 1000) offer her some stamps. In a second line you’ll get the number of stamps (from 1 to 10000) each of her friends id offering.

Output

The output for every scenario begins with a line containing “Scenario #i:”, where i is the number of the scenario starting at 1. Then print a single line with the minimum number of friends Lucy needs to borrow stamps from. If it’s impossible even if she borrows everything from everybody, write impossible. Terminate the output for the scenario with a blank line.

Example

Input:

```
3
100 6
13 17 42 9 23 57
99 6
13 17 42 9 23 57
1000 3
314 159 265
```

Output:

```
Scenario #1:
3

Scenario #2:
2

Scenario #3:
impossible
```

Problem – 6

Army Strength

Problem code: ARMY

The next MechaGodzilla invasion is on its way to Earth. And once again, Earth will be the battleground for an epic war.

MechaGodzilla's army consists of many nasty alien monsters, such as Space Godzilla, King Gidorah, and MechaGodzilla herself.

To stop them and defend Earth, Godzilla and her friends are preparing for the battle.

Problem specification

Each army consists of many different monsters. Each monster has a strength that can be described by a positive integer. (The larger the value, the stronger the monster.)

The war will consist of a series of battles. In each battle, the weakest of all the monsters that are still alive is killed.

If there are several weakest monsters, but all of them in the same army, one of them is killed at random. If both armies have at least one of the weakest monsters, a random weakest monster of MechaGodzilla's army is killed.

The war is over if in one of the armies all monsters are dead. The dead army lost, the other one won.

You are given the strengths of all the monsters. Find out who wins the war.

Input specification

The first line of the input file contains an integer T specifying the number of test cases. Each test case is preceded by a blank line.

Each test case starts with line containing two positive integers NG and NM – the number of monsters in Godzilla's and in MechaGodzilla's army. Two lines follow. The first one contains NG positive integers – the strengths of the monsters in Godzilla's army. Similarly, the second one contains NM positive integers – the strengths of the monsters in MechaGodzilla's army.

Output specification

For each test case, output a single line with a string that describes the outcome of the battle.

If it is sure that Godzilla's army wins, output the string "Godzilla".

If it is sure that MechaGodzilla's army wins, output the string "MechaGodzilla".

Otherwise, output the string "uncertain".

Example

input:

```
2
```

```
1 1
1
1

3 2
1 3 2
5 5
```

output:

```
Godzilla
MechaGodzilla
```

Hint

In the first test case, there are only two monsters, and they are equally strong. In this situation, MechaGodzilla's monster is killed and the war ends.

In the second test case, the war will consist of three battles, and in each of them one of Godzilla's monsters dies.

For all the test cases, **int** in C/C++/Java or **longint** in Pascal is enough.

Problem – 7

Will it ever Stop

Problem code: WILLITST

When Bob was in library in University of Warsaw he saw on one of facades caption :"Will it ever stop?" and below some mysterious code:

```
while n > 1
  if n mod 2 = 0 then
    n:=n/2
  else
    n:=3*n+3

Help him finding it out !
```

Input

In first line one number n<=10^14.

Output

Print "TAK" if program will stop, otherwise print "NIE"

Example

Input:

```
4
```

Output:

```
TAK
```

Problem - 8

Build a Fence

Problem code: Fence1

There is a wall in your backyard. It is so long that you can’t see its endpoints. You want to build a fence of length *L* such that the area enclosed between the wall and the fence is maximized. The fence can be of arbitrary shape, but only its two endpoints may touch the wall.

Input

The input consists of several test cases.

For every test case, there is only one integer *L* (1<=*L*<=100), indicating the length of the fence.

The input ends with *L*=0.

Output

For each test case, output one line containing the largest area. Your answer should be rounded to 2 digits after the decimal point.

Example

```
Input:
1
0

Output:
0.16
```

Problem - 9

Street Parade

Problem code: STPAR

For sure, the love mobiles will roll again on this summer's street parade. Each year, the organisers decide on a fixed order for the decorated trucks. Experience taught them to keep free a side street to be able to bring the trucks into order.

The side street is so narrow that no two cars can pass each other. Thus, the love mobile that enters the side street last must necessarily leave the side street first. Because the trucks and the ravers move up closely, a truck cannot drive back and re-enter the side street or the approach street.

You are given the order in which the love mobiles arrive. Write a program that decides if the love mobiles can be brought into the order that the organisers want them to be.

Input

There are several test cases. The first line of each test case contains a single number n, the number of love mobiles. The second line contains the numbers 1 to n in an arbitrary order. All the numbers are separated by single spaces. These numbers indicate the order in which the trucks arrive in the approach street. No more than 1000 love mobiles participate in the street parade. Input ends with number 0.

Output

For each test case your program has to output a line containing a single word `yes` if the love mobiles can be re-ordered with the help of the side street, and a single word `no` in the opposite case.

Example

Sample input:

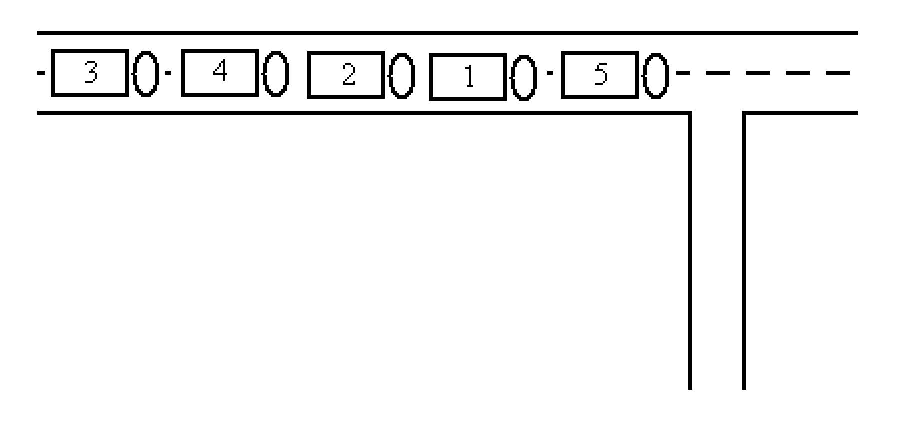
```
5
5 1 2 4 3
0
```

Sample output:

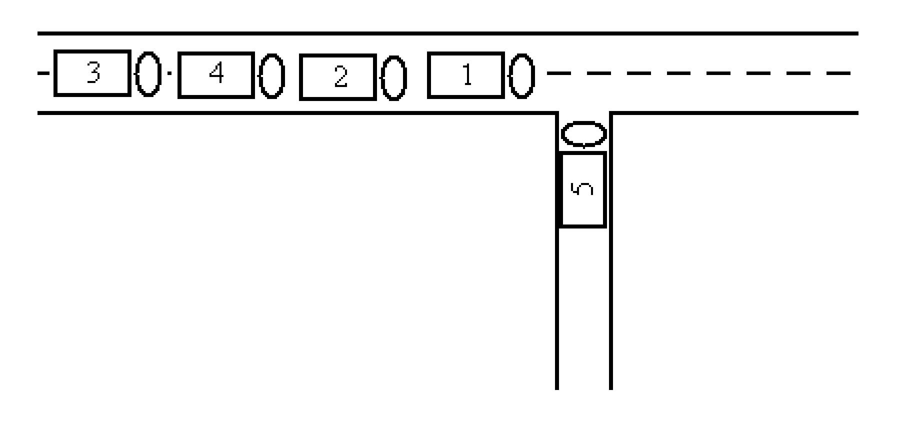
```
yes
```

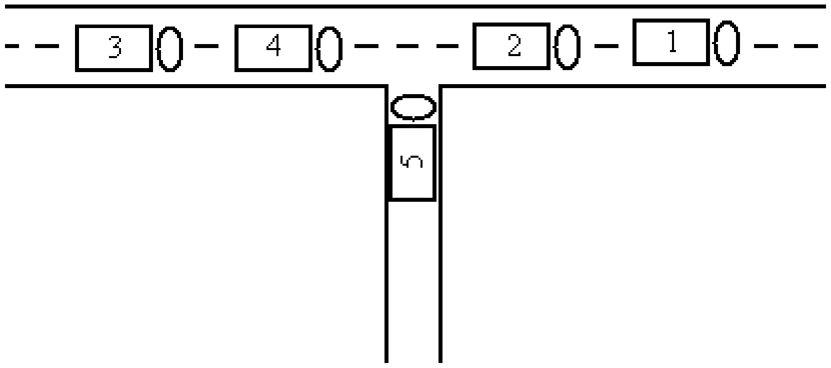
Illustration

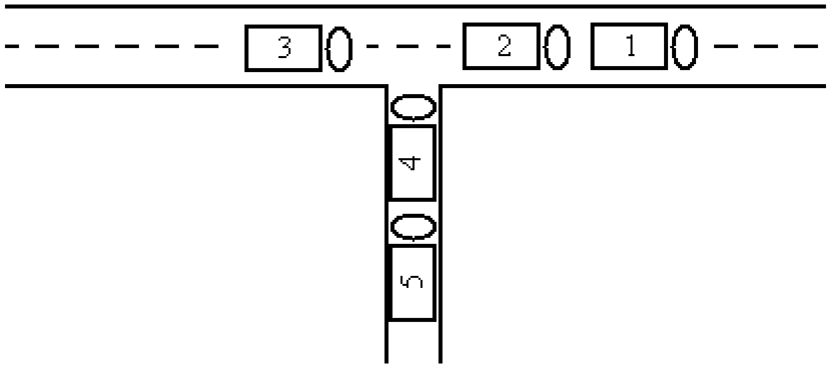
The sample input reflects the following situation:

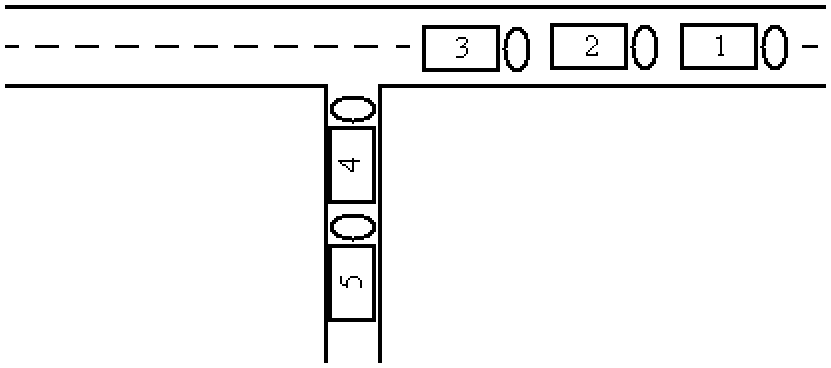


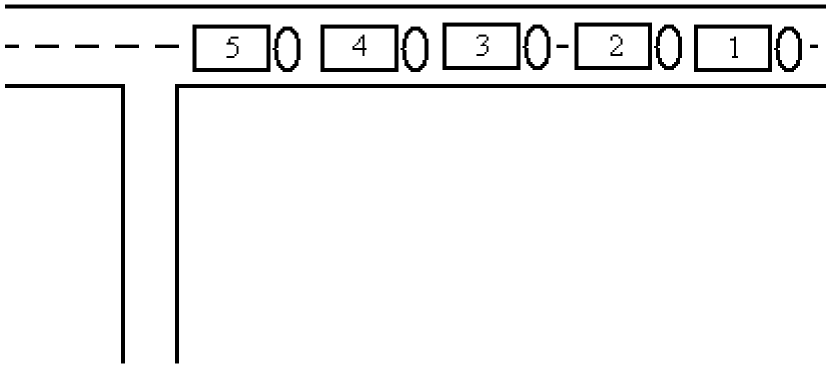
The five trucks can be re-ordered in the following way:




3
4
2
1
5


3
2
1
4
5


3
2
1
4
5


5
4
3
2
1

Problem - 10

Is it a tree

Problem code: PT07Y

You are given an unweighted, undirected graph. Write a program to check if it's a tree topology.

Input

The first line of the input file contains two integers N and M --- number of nodes and number of edges in the graph ($0 < N <= 10000$, $0 <= M <= 20000$). Next M lines contain M edges of that graph --- Each line contains a pair (u, v) means there is an edge between node u and node v ($1 <= u,v <= N$).

Output

Print *YES* if the given graph is a tree, otherwise print *NO*.

Example

Input:

```
3 2
1 2
2 3
```

Output:

```
YES
```