

## Problem A. Social Distancing

Time limit: 5 seconds

Nowadays, the Kingdom of Dreamgrid is suffering from a national pandemic. Fortunately, president Baobao is working effectively with the Center for Disease Control (CDC) and they are trying their best to make everything under control.

President Baobao has announced a policy of Social Distancing to prevent the diffusion of the virus. As the chief of CDC, you are required to research on the following problem:

There are  $n$  people who need to be observed and you have already set a monitor in  $(0, 0)$  on a 2-dimensional plane. Everyone should stay within the distance of  $r$  to the monitor. You also have to keep them stay away from each other as far as possible. To simplify the problem, you can only allocate them to integers coordinates.

Please maximize

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n d(i, j)^2,$$

where  $d(i, j)$  means the Euclidean distance between the  $i$ -th and the  $j$ -th person.

### Input

There are multiple test cases. The first line of the input contains an integer  $T$  ( $1 \leq T \leq 250$ ), indicating the number of test cases.

For each test case, the only line contains two integers  $n, r$  ( $1 \leq n \leq 8, 1 \leq r \leq 30$ ).

### Output

Please output the answer in one line for each test case.

#### Sample Input 1

```
2
4 2
5 10
```

#### Sample Output 1

```
64
2496
```

## Problem B. Mask Allocation

Time limit: 1 second

Nowadays, the Kingdom of Dreamgrid is suffering from a national pandemic. Fortunately, president Baobao is working effectively with the Center for Disease Control (CDC) and they are trying their best to make everything under control.

President Baobao has received  $n \times m$  medical masks from his friend Reku, an extremely rich billionaire. As the chief of the CDC, you are required to allocate these masks properly. There are 2 kinds of hospitals in the Kingdom of Dreamgrid,  $n$  senior hospitals for critically ill patients and  $m$  mobile cabin hospitals for patients with mild symptoms.

Before allocating them to hospitals, you have to pack them into boxes. Please note that boxes must not be opened in order to prevent contamination and you only know these boxes will be allocated to either senior hospitals or mobile cabin hospitals. That is to say, there should be a way of dividing masks boxes into  $m$  groups of  $n$  masks, and a way of dividing into  $n$  groups of  $m$  masks.

You want the number of boxes to be minimal and please also provide a lexicographically greatest sequence of the numbers of masks in boxes. We say a sequence  $a$  is lexicographically greater than another  $b$  of the same length if there exists an integer  $i$ , such that

- $a_j = b_j$ , for all  $j < i$ ; and
- $a_i > b_i$ .

### Input

There are multiple test cases. The first line of the input contains an integer  $T$  ( $1 \leq T \leq 100$ ), indicating the number of test cases.

For each test case, the only line of the input contains two integers  $n, m$  ( $1 \leq n, m \leq 10^4$ ), representing the numbers of senior hospitals and mobile cabin hospitals.

### Output

For each test case, please output two lines. The first line should contain a single integer  $k$  in the first line, indicating the minimum number of boxes. In the second line, please output  $k$  integers, denoting the lexicographically greatest sequence.

#### Sample Input 1

```
2
5 4
3 3
```

#### Sample Output 1

```
8
4 4 4 4 1 1 1 1
3
3 3 3
```

## Problem C. A National Pandemic

Time limit: 1 second

Nowadays, the Kingdom of Dreamgrid is suffering from a national pandemic. Fortunately, president Baobao is working effectively with the Center for Disease Control (CDC) and they are trying their best to make everything under control.

As the chief of CDC, you are required to update the situation simultaneously and answer queries from president Baobao himself and reporters from News agencies in the world.

Specifically, let's consider the country as a map of  $n$  nodes, denoting cities. Due to the severe pandemic, most roads and highways are closed except  $n - 1$  roads, which keep the country connected. We also define  $F(x)$  for each city  $x$  as the severity of the local disease situation. Now you need to deal with three kinds of events/queries:

1. Another outbreak happens in city  $x$  with the severity value of  $w$ . For each city  $y$ ,  $F(y)$  will be increased by  $w - \text{dist}(x, y)$ , where  $\text{dist}(x, y)$  is the number of edges on the path from  $x$  to  $y$ .
2. Update  $F(x)$  of city  $x$  to  $\min(F(x), 0)$ , which means the situation in city  $x$  has been improved with the effort of the government and medical faculties.
3. Query for the severity value  $F(x)$  of city  $x$ .

Every city's severity value  $F$  is 0 initially. Also,  $F(x)$  can be negative in some moment and we don't have to adjust it.

### Input

There are multiple test cases. The first line of the input contains an integer  $T$  ( $1 \leq T \leq 5$ ) indicating the number of test cases.

For each test case, the first line contains two integers  $n, m$  ( $1 \leq n, m \leq 5 \times 10^4$ ), representing the number of cities and the number of events and queries. The following  $n - 1$  lines describe all paths in the country, each of which contains two integers  $x, y$  ( $1 \leq x, y \leq n$ ), representing a road between city  $x$  and  $y$ . The following  $m$  lines describe all events, each starting with an integer  $opt$  ( $1 \leq opt \leq 3$ ), and if  $opt$  is

1. there will be two integers  $x, w$  ( $1 \leq x \leq n, 0 \leq w \leq 10000$ ) in the same line. This refers to Event 1 in the description above.
2. there will an integer  $x$  ( $1 \leq x \leq n$ ) in the same line. This refers to event 2.
3. there will an integer  $x$  ( $1 \leq x \leq n$ ) in the same line. This refers to the query you need to reply.

### Output

Output an integer for each query in one line.

**Sample Input 1**

```
1
5 6
1 2
1 3
2 4
2 5
1 1 5
3 4
2 1
1 2 7
3 3
3 1
```

**Sample Output 1**

```
3
9
6
```

## Problem D. Fake News

Time limit: 1 second

McDonald Thumb, the greatest president ever in the history of the Great Tokitsukaze Kingdom has held his 100th press conference about the global pandemic after making his 1000000th tweets with his smartphone. With a big smile on his face, Mr. Thumb said that nobody knows math more than he does.

‘I learn math since I was born and I am very good at it’, said Mr. Thumb. ‘People keep asking me why I know math so much and I sometimes find myself having those amazing ideas about math.’

‘For example?’, asked by a reporter.

‘Oh you, I know you! Fake news! You and your agency always produce fake news!’, replied angrily by Mr. Thumb. ‘Let me teach you something, do you know if you add up the squares of numbers from 1 to  $n$ , the sum will never be a square number?’

As another reporter in that press conference, you also wondering that given  $n$ , whether  $\sum_{k=1}^n k^2$  is a square number. Specifically, we regard a positive number  $x$  as a square number when there exists an integer  $y$  satisfying  $y^2 = x$ .

### Input

There are multiple test cases. The first line of the input contains an integer  $T$  ( $1 \leq T \leq 10^6$ ) indicating the number of test cases.

For each test case, the first line contains one integer  $n$  ( $1 \leq n \leq 10^{15}$ ).

### Output

If the sum is a square number, please output Fake news! in one line. Otherwise, please output Nobody knows it better than me! instead.

#### Sample Input 1

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

#### Sample Output 1

```
Fake news!
Nobody knows it better than me!
Nobody knows it better than me!
Nobody knows it better than me!
Nobody knows it better than me!
```

## Problem E. NeoMole Synthesis

Time limit: 1 second

The chemists have just conceived a large category of molecules, named NeoMole. The exceptionally outstanding properties of these molecules make them potential medicine for the current influenzal pneumonia.

Theoretically speaking, a NeoMole molecule consists of only one chemical element, neonium. The neonium atoms are connected by regular chemical bonds in a NeoMole molecule. Unlike carbon, the number of atoms connected to a neonium atom by chemical bonds is, surprisingly, unlimited. Also, the molecule structure is acyclic.

The scientists want to synthesize some NeoMoles, using small NeoMole molecules as substrates (i.e., raw materials). To synthesize big molecules, one can choose two molecules, which can be either substrates or previously synthesized molecules, and specify an atom for each molecule. After some mysterious chemical reactions, a new chemical bond will form between the specified atoms.

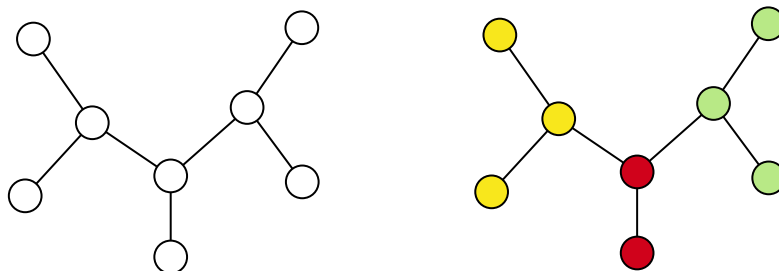


Figure 1: An optimal synthesis plan for sample test case 1.

Because the NeoMoles are really precious, scientists want to minimize their total cost on substrates. Can you write a program to help them find the optimal synthesis plan? Do not miss the opportunity to earn big money!

### Input

The first line contains a integer  $n$  ( $1 \leq n \leq 500$ ), denoting the number of atoms in the molecule to synthesize. The atoms are numbered 1 through  $n$ . This is followed by  $n - 1$  lines, each containing two integer  $u, v$  ( $1 \leq u, v \leq n$ ), denoting a chemical bond between the  $u$ -th and the  $v$ -th atoms.

Then follows a line containing a single integer  $m$  ( $1 \leq m \leq 200$ ), denoting the number of substrates. The remaining part of the input specifies these substrate molecules. The specification of each substrate molecule begins with a line containing two integers  $n, c$  ( $1 \leq n \leq 500, 1 \leq c \leq 10^6$ ), denoting the number of atoms in the molecule and the cost of the molecule per copy. It is then followed by  $n - 1$  lines, specifying the bonds between molecules, in the same way as above. It is guaranteed that the sum of the numbers of atoms in all substrate molecules does not exceed 500. You may use arbitrarily many copies of any substrate molecule in your synthesis.

It is guaranteed that, in all molecules, the atoms are connected acyclicly.

### Output

Output a single line of an integer, denoting the minimum total cost on raw materials. If it is impossible to synthesize the target molecule, output `impossible` instead.

**Sample Input 1**

```
8
1 2
2 3
2 4
4 5
4 6
6 7
6 8
3
3 2
1 2
2 3
2 3
1 2
1 5
```

**Sample Output 1**

```
7
```

**Sample Input 2**

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

**Sample Output 2**

```
impossible
```

## Problem F. Tokens on the Tree

Time limit: 1 second

Chiaki has a tree with  $n$  vertices. There might be a token colored white or black on each vertex of the tree, and there are exactly  $w$  white tokens and exactly  $b$  black tokens. Also, for each pair of vertices with the same color of tokens, there must have been a path between them such that every vertex on the path contains a token of the same color.

Chiaki would like to perform the following operations:

1. Choose a vertex  $u$  with a token.
2. Choose a path  $p_1, p_2, \dots, p_k$  such that  $p_1 = u$ , all vertices  $p_1, p_2, \dots, p_{k-1}$  contain a token of the same color, and there's no token in  $p_k$ .
3. Move the token in  $p_1$  to  $p_k$ . Now there's no token in  $p_1$  and  $p_k$  contains a token.

For two initial configurations of tokens  $S$  and  $T$ , if Chiaki could perform the above operations several times to make  $S$  become  $T$ ,  $S$  and  $T$  are considered equivalent.

Let  $f(w, b)$  be the number of equivalence classes (an equivalence class is a maximal set of configurations where each two of them are equivalent; all equivalence classes partition the set of all configurations), Chiaki would like to know the value of

$$\left( \sum_{w=1}^{n-1} \sum_{b=1}^{n-w} w \cdot b \cdot f(w, b) \right) \bmod (10^9 + 7).$$

### Input

There are multiple test cases. The first line of input contains an integer  $T$ , indicating the number of test cases.

For each test case, the first line contains an integer  $n$  ( $2 \leq n \leq 2 \times 10^5$ ) – the number of vertices in the tree. The second line contains  $n - 1$  integers  $p_2, p_3, \dots, p_n$  ( $1 \leq p_i < i$ ), where  $p_i$  means there is an edge between vertex  $i$  and vertex  $p_i$ .

It's guaranteed that the sum of  $n$  of all test cases will not exceed  $2 \times 10^5$ .

### Output

For each test case, output an integer denoting the value of

$$\left( \sum_{w=1}^{n-1} \sum_{b=1}^{n-w} w \cdot b \cdot f(w, b) \right) \bmod (10^9 + 7).$$

#### Sample Input 1

```
2
5
1 2 3 3
10
1 2 3 4 3 6 3 8 2
```

#### Sample Output 1

```
71
989
```

### Note

For the first sample, the value of  $f(w, b)$  for each  $w$  and  $b$ :  $f(1, 1) = 1$ ,  $f(1, 2) = 2$ ,  $f(1, 3) = 3$ ,  $f(1, 4) = 3$ ,  $f(2, 1) = 2$ ,  $f(2, 2) = 2$ ,  $f(2, 3) = 1$ ,  $f(3, 1) = 3$ ,  $f(3, 2) = 1$ , and  $f(4, 1) = 3$ .



## Problem G. Topo Counting

Time limit: 4 seconds

Now we start to describe a kind of directed graph called the Drying Rack Graph (DRG) with a parameter  $N$ .

A DRG contains  $N$  groups of vertexes. The  $i$ -th group  $V^i$  contains  $2N$  vertices:  $V_1^i, V_2^i, \dots, V_{2N}^i$ .

There are two types of edges in DRG: intra-group edges (edges inside each group) and inter-group edges (edges between groups).

**Intra-Group Edge** For the  $i$ -th group, the following intra-group edges exist:

- $(V_j^i, V_{j+N}^i)$ , for all integer  $j$  such that  $1 \leq j \leq N$ ;
- $(V_j^i, V_{j+1}^i)$ , for all integer  $j$  such that  $1 \leq j \leq N-1$  or  $N+1 \leq j \leq 2N-1$ .

**Inter-Group Edge** The following inter-group edges exist:

- $(V_{i+N}^1, V_1^{i+1})$ , for all integer  $i$  such that  $1 \leq i \leq N-1$ ;
- $(V_i^1, V_{1+N}^i)$ , for all integer  $i$  such that  $2 \leq i \leq N$ .

Now we want to know the number of topo-order of a DRG parameterized with  $N$ .

A topo-order of a directed graph  $G = (V, E)$  is a permutation  $v_{p_1}, v_{p_2}, \dots, v_{p_{|V(G)|}}$  of all vertices from  $V(G)$  such that for all  $i < j$ ,  $(v_{p_j}, v_{p_i}) \notin E(G)$

In order to avoid calculations of huge integers, report answer modulo a prime  $M$  instead.

### Input

The input contains only two integers  $N, M$  ( $1 \leq N \leq 5000, 2 * N * N < M \leq 2^{30}$ ), and  $M$  is a prime.

### Output

Output one integer indicating the answer.

Sample Input 1	Sample Output 1
2 1073741789	31
Sample Input 2	Sample Output 2
3 1073741789	7954100

## Problem H. Dividing

Time limit: 1 second

The following rules define a kind of integer tuple - the Legend Tuple:

- $(1, k)$  is always a Legend Tuple, where  $k$  is an integer.
- if  $(n, k)$  is a Legend Tuple,  $(n + k, k)$  is also a Legend Tuple.
- if  $(n, k)$  is a Legend Tuple,  $(nk, k)$  is also a Legend Tuple.

We want to know the number of the Legend Tuples  $(n, k)$  where  $1 \leq n \leq N, 1 \leq k \leq K$ .

In order to avoid calculations of huge integers, report the answer modulo  $10^9 + 7$  instead.

### Input

The input contains two integers  $N$  and  $K$ ,  $1 \leq N, K \leq 10^{12}$ .

### Output

Output the answer modulo  $10^9 + 7$ .

#### Sample Input 1

3 3

#### Sample Output 1

8

#### Sample Input 2

3 9

#### Sample Output 2

14

## Problem I. Valuable Forests

Time limit: 6 seconds

We define the value of an unrooted tree  $T$  as  $\sum_{u \in V(T)} (d(u))^2$ , where  $V(T)$  is the set of all vertices of  $T$ , and  $d(u)$  is the degree of the vertex  $u$ . We define the value of a forest as the sum of the value of all trees from it. Now we want you to answer the sum of the value of all forests with  $N$  labeled vertices.

In order to avoid calculations of huge integers, report answer modulo a prime  $M$  instead.

### Input

There are multiple test cases. The first line of the input contains two integers  $T$  and  $M$  ( $1 \leq T \leq 3000, 1 \leq M \leq 2^{30}$ , and  $M$  is a prime), indicating the number of test cases and the modulus.

For each test case, the only line contains an only integer  $N$  ( $1 \leq N \leq 5000$ ).

### Output

For each test case, output the sum of answer modulo  $M$  in one line.

#### Sample Input 1

```
5 1000000007
2
3
4
5
107
```

#### Sample Output 1

```
2
24
264
3240
736935633
```

## Problem J. Pointer Analysis

Time limit: 1 second

Pointer analysis, which aims to figure out which objects accessible via a specific pointer variable in a program during the execution, is one of the fundamental parts of static program analysis. Now we want you to perform the context-insensitive pointer analysis on the test data.

A program contains 26 objects denoted by lowercase letters and each object has 26 member variables (a.k.a. fields, which are pointers that may point to some objects) denoted by lowercase letters as well. Meanwhile, there are 26 global pointers in the programs designated by uppercase letters.

There are four kinds of statements in a program. We use `[Variable]` to represent the name of a pointer, `[Field]` to represent the name of a member variable, and `[Object]` to represent an object.

	Format	Example	Description
Allocation	<code>[Variable] = [Object]</code>	$A = x$	pointer $A$ can point to object $x$ (i.e., $x$ is accessible via $A$ )
Assignment	<code>[Variable] = [Variable]</code>	$A = B$	pointer $A$ can point to every object accessible via $B$
Store	<code>[Variable].[Field] = [Variable]</code>	$A.f = B$	for every object $o$ accessible via $A$ , the member variable $f$ of $o$ can point to every object accessible via $B$
Load	<code>[Variable] = [Variable].[Field]</code>	$A = B.f$	for every object $o$ accessible via $B$ , $A$ can point to every object accessible via the member variable $f$ of $o$

The context-insensitive pointer analysis assumes that statements of the program will be executed in any order for a sufficient number of times. For example, in the following two programs, both  $A$  and  $B$  can point to the object  $x$  and object  $o$ . The reason for that is, in the real world, the exact execution order and execution times of statements are difficult to predict.

First Program	Second Program
$A = o$	$B = A$
$A = x$	$A = x$
$B = A$	$A = o$

Now you are asked to perform a context-insensitive pointer analysis on a given program consists of  $N$  statements, and for each pointer, output the objects it can point to.

### Input

The first line of the input contains one integer  $N$  ( $1 \leq N \leq 200$ ), representing the number of statements in the program. There is exactly one space before and after the equal sign '='.

Each of the following  $N$  lines contains one statement.

### Output

The output should contains 26 lines.

In the  $i$ -th line, output the name of the  $i$ -th pointer (which is the  $i$ -th uppercase letter) followed by a colon ':' and a space, and then list the objects accessible via this pointer in alphabetical order.

**Sample Input 1**

```
5
B.f = A
C = B.f
C = x
A = o
B = o
```

**Sample Output 1**

```
A: o
B: o
C: ox
D:
E:
F:
G:
H:
I:
J:
K:
L:
M:
N:
O:
P:
Q:
R:
S:
T:
U:
V:
W:
X:
Y:
Z:
```

**Sample Input 2**

```
4
A = o
B.f = A
C = B.f
C = g
```

**Sample Output 2**

```
A: o
B:
C: g
D:
E:
F:
G:
H:
I:
J:
K:
L:
M:
N:
O:
P:
Q:
R:
S:
T:
U:
V:
W:
X:
Y:
Z:
```

**Sample Input 3**

```
3
A = o
B = A
A = x
```

**Sample Output 3**

```
A: ox
B: ox
C:
D:
E:
F:
G:
H:
I:
J:
K:
L:
M:
N:
O:
P:
Q:
R:
S:
T:
U:
V:
W:
X:
Y:
Z:
```