标题：A、Singing Contest | 时间限制：1 秒 | 内存限制：256M

Jigglypuff is holding a singing contest. There are 2^n singers indexed from 1 to 2^n participating in the contest.

The rule of this contest is like the knockout match. That is, in the first round, singer 1 competes with singer 2, singer 3 competes with singer 4 and so on; in the second round, the winner of singer 1 and singer 2 competes with the winner of singer 3 and singer 4 and so on. There are n rounds in total.

Each singer has prepared n songs before the contest. Each song has a unique pleasantness. In each round, a singer should sing a song among the songs he prepared. In order not to disappoint the audience, one song cannot be performed more than once. The singer who sings the song with higher pleasantness wins.

Now all the singers know the pleasantness of songs prepared by all the others. Everyone wants to win as many rounds as he can. Assuming that singers choose their song optimally, Jigglypuff wants to know which singer will win the contest?

输入描述:

The input starts with one line containing exactly one integer t which is the number of test cases. (1 ≤ t ≤ 10)

For each test case, the first line contains exactly one integer n where 2^n is the number of singers. (1 ≤ n ≤ 14)

Each of the next 2^n lines contains n integers where a_{ij} is the pleasantness of the j-th song of the i-th singer. It is guaranteed that all these 2^nx n integers are pairwise distinct. (1≤ a_{ij} ≤ 10^9)

输出描述:

For each test case, output "Case #x: y" in one line (without quotes), where x is the test case number (starting from 1) and y is the index of the winner.

示例 1

输入

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

输出

```
Case #1: 2
Case #2: 4
```

标题：B、Endless Pallet | 时间限制：2 秒 | 内存限制：256M

There is a tree formed by N nodes. Initially, all the nodes are in white color. If there is at least one white node on the tree, Ash will randomly choose a path on the tree and dye all the nodes in the path black. Otherwise, Ash will stop dyeing and go home.

Given the tree, can you calculate the expected number of the paths Ash choose?

There are $\frac{N(N+1)}{2}$ paths in total, that is, path from u to v and path from v to u are the same. It is okay to choose a path with no white node, but Ash will stop dyeing immediately when there are no white nodes on the tree.

输入描述:

The input starts with one line containing exactly one integer T, which is the number of test cases.

Each test case starts with one line containing exactly one integer N, indicating the size of the tree.

Then followed by N lines, each consists of 2 numbers u_i, v_i, indicating the i-th edge of the tree is between u_i and v_i.

- $1 \leqslant T \leqslant 15$.
- $1 \leqslant N \leqslant 50$.
- $1 \leqslant u_i, v_i \leqslant n$.

输出描述:

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 number of the paths Ash choose.

In order to avoid floating point arithmetic, you are supposed to output y modulo 998244353, that means if the answer is equal to $\frac{P}{Q}$, you should output $y = P \cdot Q^{-1} \mod 998244353$

示例 1

输入

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

输出

```
Case #1: 1
Case #2: 2
Case #3: 698771050
```

标题：C、Generation I | 时间限制：3 秒 | 内存限制：256M

Oak is given N empty and non-repeatable sets which are numbered from 1 to N.

Now Oak is going to do N operations. In the i-th operation, he will insert an integer x between 1 and M to every set indexed between i and N.

Oak wonders how many different results he can make after the N operations. Two results are different if and only if there exists a set in one result different from the set with the same index in another result.

Please help Oak calculate the answer. As the answer can be extremely large, output it modulo 998244353.

输入描述:

The input starts with one line containing exactly one integer T which is the number of test cases. (1 ≤ T ≤ 20)

Each test case contains one line with two integers N and M indicating the number of sets and the range of integers. (1 ≤ N ≤ 10^{18}, 1 ≤ M ≤ 10^{18}, $1 \leq \min\{N, M\} \leq 10^6$)

输出描述:

For each test case, output "Case #x: y" in one line (without quotes), where x is the test case number (starting from 1) and y is the number of different results modulo 998244353.

示例 1

输入

```
2
2 2
3 4
```

输出

```
Case #1: 4
Case #2: 52
```

标题：D、Bulbasaur | 时间限制：1 秒 | 内存限制：256M

Silph company deployed a passenger flow analysis system in a clothing store that captures photos of human faces and photos of human bodies in real time.

In order to analyze the passenger flow better, the system will associate human faces with human bodies. In other words, there are some edges connecting faces with bodies. Each edge has a positive weight.

However, due to lack of precision and accuracy of the algorithms provided by this company, these associations may not be completely correct.

In a correct relationship, one human face can associate with multiple human bodies (one person may change multiple suits of clothes), but one human body cannot associate with multiple human faces.

Now Bulbasaur works as an intern at Silph company and the boss asks him to solve this problem.

Bulbasaur is supposed to find an association relationship, such that the sum of weights of the association edges is maximum.

输入描述:

The input starts with one line containing exactly one integer T, which is the number of test cases.

For each test case, the first line contains three integers n, m and k, indicating the number of face photos, the number of body photos, and the number of existing association edges, respectively.

Then followed by k lines, each consists of three integers a_i, b_i and c_i, representing an edge weighted c_i connecting the a_i-th face photo with the b_i-th body photo.

- $1 \leqslant T \leqslant 5$.
- $1 \leqslant n, m, k \leqslant 10^5$.
- $1 \leqslant a_i \leqslant n$.
- $1 \leqslant b_i \leqslant m$.
- $1 \leqslant c_i \leqslant 10^9$.
- $\forall i \neq j \rightarrow (a_i \neq a_j \vee b_i \neq b_j)$.

输出描述:

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 sum of weights of the association edges.

示例 1

输入

```
1
2 3 3
1 2 1919
1 3 810
2 2 450
```

输出

Case #1: 2729

标题：E、Charmander | 时间限制：4 秒 | 内存限制：256M

Charmander has a magical string s whose length is n.

At each second, every character in string s expands simultaneously, where character i will become the string S_i. That means if the string contains 3 characters c_1, c_2 and c_3 , in next second the string will become $S_{c_1} + S_{c_2} + S_{c_3}$.

But at any moment, each character that appears in string s can only be one of the m characters numbered from 1 to m.

Given a target string t, Charmander wants to know in which second it first appears as a substring of string s, or if it never appears?

输入描述:

The input starts with one line containing exactly one integer T, which is the number of test cases.

For each test case, the first line contains three integers n, m and k, indicating the length of string s in second 0, the size of character set and the length of string t.

Then followed by one line, consisting of n integers, indicating the string s in second 0.

Then followed by m lines, each consists of k_i, $S_i[1]$, ••• , $S_i[k_i]$, representing the string S_i.

Then followed by one line, consisting of k integers, indicating the string t.

- $1 \leqslant T \leqslant 10$.
- $1 \leqslant n,m,k \leqslant 1000$.
- $1 \le \sum_{i=1}^{m} k_i \le 1000$
- $1 \leqslant s[i], S_i[j], t[i] \leqslant m$.
- $k_i \geqslant 2$.

输出描述:

For each test case, output one line containing "Case #x: y", where x is the test case number (starting from 1) and string t first appears in second y.

If string t never appears, y is supposed to be -1.

示例 1

输入

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

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

输出

```
Case #1: 1
Case #2: 2
```

标题：F、Squirtle | 时间限制：3 秒 | 内存限制：256M

Squirtle has a binary expression tree which is rooted at node 1.

There are n leaves on the tree, each leaf of which contains a binary number that can be either 0 or 1, and each non-leaf node contains a binary logical operator.

There are 16 types of binary logical operator in total. Here we use an integer from 0 to 15 to represent a binary logical operator. Suppose that an integer i's binary expression is $\overline{i_3 i_2 i_1 i_0}$, then it represents $f_i : \{0,1\}^2 \to \{0,1\}$ where $f_i(0,0)=i_0$,$f_i(0,1)=i_1$,$f_i(1,0)=i_2$,$f_i(1,1)=i_3$. The first parameter corresponds to the left operand and vice versa. For example, the bitwise-and operator is represented by 8.

Squirtle can fill non-leaf node i with a binary logical operator in set S_i.

Given the tree and all the sets, Squirtle wants to know how to fill non-leaf nodes so that the sum of the results of all the expressions, where the leaves take all the possible values, is maximum.

输入描述:

The input starts with one line containing exactly one integer t which is the number of test cases. (1 ≤ t ≤ 20)

For each test case, the first line contains exactly one integer n which is the number of leaf nodes. (2 ≤ n ≤ 2000)

Each of the next n-1 lines contains a string s_i of length 16. It is guaranteed that s_i only consists of 0 and 1 and contains at least one 1. If s_i's j-th character(numbered from 0) s_{ij}=1, then operator j∈ S_i, otherwise operator $j \notin S_i$.

Each of the next 2n-2 lines contains an integer a_i, which represents the father of i+1. It is guaranteed that the tree is valid. That is, the tree is a binary tree whose nodes have either 0 or 2 children. Nodes from 1 to n-1 are guaranteed to be non-leaf nodes. The child with the smaller index is regarded as the left operand. (1 ≤ a_i ≤ i)

输出描述:

For each test case, output "Case #x: y" in one line (without quotes), where x is the test case number (starting from 1) and y is the maximum possible sum.

示例 1

输入

```
2
2
0000000010000010
1
1
3
0000000010000010
0000000010000011
1
1
```

```
2
2
```

输出

```
Case #1: 3
Case #2: 8
```

标题：G、Pikachu | 时间限制：3 秒 | 内存限制：256M

In Viridian forest there is a tree T formed by N nodes, each edge on which has a positive weight.

There is an undirected graph G generated from tree T, which contains N nodes and $\binom{N}{2}$ undirected edges, where the capacity of the edge between u and v equals to the distance between them on the tree T.

Given the tree T, Pikachu wants to calculate the sum of the max flow between every two nodes in G, there are $\binom{N}{2}$ different pairs of nodes should be counted. Could you help him?

输入描述:

The input starts with one line containing exactly one integer t, which is the number of test cases.

For each test case, the first line contains one integer N, indicating the size of the tree T.

Then followed by N - 1 lines, each consists of three integers u_i, v_i and w_i ,representing the two nodes connected by the i-th edge and the weight of the i-th edge.

- $1 \leqslant t \leqslant 10$.
- $2 \leqslant N \leqslant 10^5$.
- $1 \leqslant w_i \leqslant 1000$.
- $1 \leq u_i \neq v_i \leq N$.

输出描述:

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 sum of the maximum flow between every two nodes in G.

示例 1

输入

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

输出

```
Case #1: 7
Case #2: 72
```

标题：H、Eevee | 时间限制：3 秒 | 内存限制：256M

One day, Eevee received an equation from her best friend:

a^x x b^y = c^z,

where c and z are given integers.

She wants to count the number of integral solutions (a,x,b,y) of this equation which satisfies 1⩽ a,b ⩽ m, 0 ⩽ x,y ⩽ m.

输入描述:

The input starts with one line containing exactly one integer T which is the number of test cases. (1 ⩽ T⩽ 1000)

Each test case contains one line with three integers c, z and m. (1 ⩽ c, m ⩽ 10^5,0 ⩽ z ⩽ 10^5)

输出描述:

For each test case, output "Case #x: y" in one line (without quotes), where x is the test case number (starting from 1) and y is the number of integral solutions.

示例 1

输入

```
2
2 2 3
6 2 36
```

输出

```
Case #1: 13
Case #2: 301
```

标题：I、Team Rocket | 时间限制：4 秒 | 内存限制：256M

There are n trains running between Kanto and Johto region. Assuming the railway is a number line, the i-th train travels from coordinate l_i to coordinate r_i (both inclusive).
One day, m Team Rocket members invaded the railway system successfully. The i-th Team Rocket member was going to destroy the transportation hub with coordinate x_i. Once a transportation hub within the driving range of a train is destroyed, the train's itinerary will be canceled immediately. Giovanni wants to know how many train trips will be firstly canceled after each attack.
After all the attacks finished, for each train Giovanni needs to know that in which attack its itinerary was firstly canceled, or it was unaffected at all.

输入描述:
The input starts with one line containing exactly one integer T, which is the number of test cases.

For each test case, the first line contains two integers n and m, indicating the number of trains and the number of Team Rocket members.

Each of the next n lines contains 2 integers l_i and r_i, indicating the driving range of the i-th train.

Each of the next m lines contains exactly one integer $y_i = (x_i \oplus (res_{i-1} \mod 998244353))$, where x_i is the transportation hub that Team Rocket members would destroy in the i-th attack, res_{i-1} is the product of the indexes of trips cancelled by the (i-1)-th attack and $\oplus$ means exclusive or.

If no such trip exists, res_{i-1} is considered to be 0.

- $1 \leqslant T \leqslant 5$.
- $1 \leqslant n,m \leqslant 2 \times 10^5$.
- $-10^9 \leqslant l_i \leqslant r_i \leqslant 10^9$.
- $-10^9 \leqslant x_i \leqslant 10^9$.

输出描述:
For each test case, output one line "Case #x:" first, where x is the test case number (starting from 1).

Then output m lines, each line of which contains exactly one integer y_i, indicating the number of train trips firstly canceled after the i-th attack.

Finally output one line, containing n integers, where the i-th integer is the time when the i-th train trip is firstly canceled or 0 if it is not affected.
示例 1
输入

```
1
3 4
1 3
```

```
2 6
-999 1000000000
-1000
1
5
2
```

输出

```
Case #1:
0
2
1
0
2 3 2
```

标题：J、Heritage of skywalkert | 时间限制：1 秒 | 内存限制：256M

skywalkert, the new legend of Beihang University ACM-ICPC Team, retired this year leaving a group of newbies again.

Rumor has it that he left a heritage when he left, and only the one who has at least 0.1% IQ(Intelligence Quotient) of his can obtain it.

To prove you have at least 0.1% IQ of skywalkert, you have to solve the following problem:

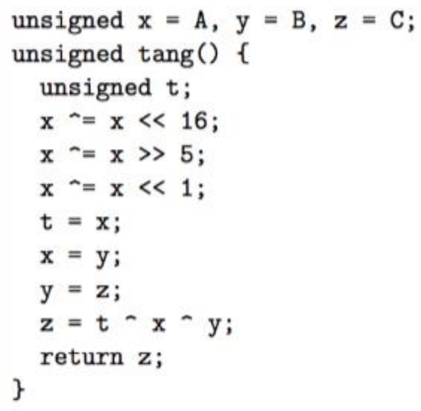
Given n positive integers, for all (i, j) where 1 ⩽ i, j ⩽ n and i ≠ j, output the maximum value among $\operatorname{lcm}(a_i, a_j)$. lcm means the Lowest Common Multiple.

输入描述:

The input starts with one line containing exactly one integer t which is the number of test cases. (1 ⩽ t ⩽ 50)

For each test case, the first line contains four integers n, A, B, C. (2 ⩽ n ⩽ 10^7, A, B, C are randomly selected in unsigned 32 bits integer range)

The n integers are obtained by calling the following function n times, the i-th result of which is a_i, and we ensure all $a_i > 0$. Please notice that for each test case x, y and z should be reset before being called.



```
unsigned x = A, y = B, z = C;
unsigned tang() {
  unsigned t;
  x ^= x << 16;
  x ^= x >> 5;
  x ^= x << 1;
  t = x;
  x = y;
  y = z;
  z = t ^ x ^ y;
  return z;
}
```

No more than 5 cases have n greater than 2 x 10^6.

输出描述:

For each test case, output "Case #x: y" in one line (without quotes), where x is the test case number (starting from 1) and y is the maximum lcm.

示例 1

输入

2

2 1 2 3

5 3 4 8

输出

Case #1: 68516050958

Case #2: 5751374352923604426