Large Division

Time Limit: 1000 ms & Memory Limit: 32768 kB

Given two integers, **a** and **b** , you should check whether **a** is divisible by **b** or not. We know that an integer **a** is divisible by an integer **b** if and only if there exists an integer **c** such that **a = b * c**.

Input

Input starts with an integer **T (≤ 525)**, denoting the number of test cases.

Each case starts with a line containing two integers **a** **($-10^{200} \le a \le 10^{200}$)** and **b (|b| > 0, b fits into a 32 bit signed integer)**. Numbers will not contain leading zeroes.

Output

For each case, print the case number first. Then print **'divisible'** if **a** is divisible by **b** . Otherwise print **'not divisible'**.

Sample Input

```
6
101 101
0 67
-101 101
7678123668327637674887634 101
11010000000000000000 256
-202202202202000202202202 -101
```

Sample Output

```
Case 1: divisible
Case 2: divisible
Case 3: divisible
Case 4: not divisible
Case 5: divisible
Case 6: divisible
```

Commandos

Time Limit: 2000 ms & Memory Limit: 32768 kB

A group of commandos were assigned a critical task. They are to destroy an enemy head quarter. The enemy head quarter consists of several buildings and the buildings are connected by roads. The commandos must visit each building and place a bomb at the base of each building. They start their mission at the base of a particular building and from there they disseminate to reach each building. The commandos must use the available roads to travel between buildings. Any of them can visit one building after another, but they must all gather at a common place when their task in done. In this problem, you will be given the description of different enemy headquarters. Your job is to determine the minimum time needed to complete the mission. Each commando takes exactly one unit of time to move between buildings. You may assume that the time required to place a bomb is negligible. Each commando can carry unlimited number of bombs and there is an unlimited supply of commando troops for the mission.

Input

Input starts with an integer **T (≤50)**, denoting the number of test cases.

The first line of each case starts with a positive integer **N (1 ≤ N ≤ 100)** , where **N** denotes the number of buildings in the head quarter. The next line contains a positive integer **R** , where **R** is the number of roads connecting two buildings. Each of the next **R** lines contain two distinct numbers **u v (0 ≤ u, v < N)** , this means there is a road connecting building **u** to building **v** . The buildings are numbered from **0** to **N- 1** . The last line of each case contains two integers **s d (0 ≤ s, d < N)** . Where **s** denotes the building from where the mission starts and **d** denotes the building where they must meet. You may assume that two buildings will be directly connected by at most one road. The input will be given such that, it will be possible to go from any building to another by using one or more roads.

Output

For each case, print the case number and the minimum time required to complete the mission.

Sample Input

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

Sample Output

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

Coprime Subsequences

Time Limit: 2000 ms & Memory Limit: 262144 kB

Let's call a non-empty sequence of positive integers a_1, a_2... a_k coprime if the greatest common divisor of all elements of this sequence is equal to 1.

Given an array a consisting of n positive integers, find the number of its coprime subsequences. Since the answer may be very large, print it modulo $10^9 + 7$.

Note that two subsequences are considered different if chosen indices are different. For example, in the array [1, 1] there are 3 different subsequences: [1], [1] and [1, 1].

Input

The first line contains one integer number n ($1 \le n \le 100000$).

The second line contains n integer numbers a_1, a_2... a_n ($1 \le a_i \le 100000$).

Output

Print the number of coprime subsequences of a modulo $10^9 + 7$.

Examples

Input

```
3
1 2 3
```

Output

```
5
```

Input

```
4
1 1 1 1
```

Output

```
15
```

Input

```
7
1 3 5 15 3 105 35
```

Output

```
100
```

Note

In the first example coprime subsequences are:

1. 1
2. 1, 2
3. 1, 3
4. 1, 2, 3
5. 2, 3

In the second example all subsequences are coprime.

Calculate your GPA

Time Limit: 1000 ms & Memory Limit: 1572864 kB

Design a calculator to calculate your GPA. Possible grades are S, A, B, C, D and E. S corresponds to 10 credits, A to 9, B to 8, C to 7, D to 6 and E to 5. There are n subjects. Each subject has a number of credits and the grade obtained by the student.
GPA of the student is the summation of [[grades obtained multiplied by the number of credits for each subject]] divided by the total number of credits. for example,
if input is n = 3 and three subjects have credits 3, 4 and 5. a student has obtained grade S, A and B respectively, then GPA = (3*10 + 4*9 + 5*8)/(3+4+5)

Input

First line consists of t, the number of test cases (1 <= t <= 20)
Each test case starts with n, the number of subjects (1 <= n <= 16)
next n lines consist of credits for that subject followed by grade obtained by the student (1 <= credits <= 10 and grade will be one among S, A, B, C, D and E)

Output

For each test case, print the GPA of the student rounded off to two decimal places.

Example

```
Input: 1
12
4 D
1 E
1 S
3 D
1 E
2 A
4 A
4 A
3 C
1 C
1 B
2 D Output: 7.41
```

Large Division

Time Limit: 1000 ms & Memory Limit: 32768 kB

Given two integers, **a** and **b** , you should check whether **a** is divisible by **b** or not. We know that an integer **a** is divisible by an integer **b** if and only if there exists an integer **c** such that **a = b * c**.

Input

Input starts with an integer **T (≤ 525)**, denoting the number of test cases.

Each case starts with a line containing two integers **a (-** $\mathbf{10^{200}} \le$ **a** $\le \mathbf{10^{200}}$ **)** and **b (|b| > 0, b fits into a 32 bit signed integer)**. Numbers will not contain leading zeroes.

Output

For each case, print the case number first. Then print **'divisible'** if **a** is divisible by **b** . Otherwise print **'not divisible'**.

Sample Input

```
6
101 101
0 67
-101 101
7678123668327637674887634 101
11010000000000000000 256
-202202202202000202202202 -101
```

Sample Output

```
Case 1: divisible
Case 2: divisible
Case 3: divisible
Case 4: not divisible
Case 5: divisible
Case 6: divisible
```

Commandos

Time Limit: 2000 ms & Memory Limit: 32768 kB

A group of commandos were assigned a critical task. They are to destroy an enemy head quarter. The enemy head quarter consists of several buildings and the buildings are connected by roads. The commandos must visit each building and place a bomb at the base of each building. They start their mission at the base of a particular building and from there they disseminate to reach each building. The commandos must use the available roads to travel between buildings. Any of them can visit one building after another, but they must all gather at a common place when their task in done. In this problem, you will be given the description of different enemy headquarters. Your job is to determine the minimum time needed to complete the mission. Each commando takes exactly one unit of time to move between buildings. You may assume that the time required to place a bomb is negligible. Each commando can carry unlimited number of bombs and there is an unlimited supply of commando troops for the mission.

Input

Input starts with an integer **T (≤50)**, denoting the number of test cases.

The first line of each case starts with a positive integer **N (1 ≤ N ≤ 100)** , where **N** denotes the number of buildings in the head quarter. The next line contains a positive integer **R** , where **R** is the number of roads connecting two buildings. Each of the next **R** lines contain two distinct numbers **u v (0 ≤ u, v < N)** , this means there is a road connecting building **u** to building **v** . The buildings are numbered from **0** to **N- 1** . The last line of each case contains two integers **s d (0 ≤ s, d < N)** . Where **s** denotes the building from where the mission starts and **d** denotes the building where they must meet. You may assume that two buildings will be directly connected by at most one road. The input will be given such that, it will be possible to go from any building to another by using one or more roads.

Output

For each case, print the case number and the minimum time required to complete the mission.

Sample Input

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

Sample Output

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

Coprime Subsequences

Time Limit: 2000 ms & Memory Limit: 262144 kB

Let's call a non-empty sequence of positive integers a_1, a_2... a_k coprime if the greatest common divisor of all elements of this sequence is equal to 1.

Given an array a consisting of n positive integers, find the number of its coprime subsequences. Since the answer may be very large, print it modulo $10^9 + 7$.

Note that two subsequences are considered different if chosen indices are different. For example, in the array [1, 1] there are 3 different subsequences: [1], [1] and [1, 1].

Input

The first line contains one integer number n ($1 \le n \le 100000$).

The second line contains n integer numbers a_1, a_2... a_n ($1 \le a_i \le 100000$).

Output

Print the number of coprime subsequences of a modulo $10^9 + 7$.

Examples

Input

```
3
1 2 3
```

Output

```
5
```

Input

```
4
1 1 1 1
```

Output

```
15
```

Input

```
7
1 3 5 15 3 105 35
```

Output

```
100
```

Note

In the first example coprime subsequences are:

1. 1
2. 1, 2
3. 1, 3
4. 1, 2, 3
5. 2, 3

In the second example all subsequences are coprime.

Calculate your GPA

Time Limit: 1000 ms & Memory Limit: 1572864 kB

Design a calculator to calculate your GPA. Possible grades are S, A, B, C, D and E. S corresponds to 10 credits, A to 9, B to 8, C to 7, D to 6 and E to 5. There are n subjects. Each subject has a number of credits and the grade obtained by the student.
GPA of the student is the summation of [[grades obtained multiplied by the number of credits for each subject]] divided by the total number of credits. for example,
if input is n = 3 and three subjects have credits 3, 4 and 5. a student has obtained grade S, A and B respectively, then GPA = (3*10 + 4*9 + 5*8)/(3+4+5)

Input

First line consists of t, the number of test cases (1 <= t <= 20)
Each test case starts with n, the number of subjects (1 <= n <= 16)
next n lines consist of credits for that subject followed by grade obtained by the student (1 <= credits <= 10 and grade will be one among S, A, B, C, D and E)

Output

For each test case, print the GPA of the student rounded off to two decimal places.

Example

```
Input: 1
12
4 D
1 E
1 S
3 D
1 E
2 A
4 A
4 A
3 C
1 C
1 B
2 D Output: 7.41
```