The 2015 ACM ASIA Region Programming Contest

Kolkata Site

Onsite Contest Problems

Sponsored by IBM

Hosted by NITTTR Kolkata

2015

acm International Collegiate Programming Contest

IBM event sponsor

26th December 2015

11 Problems

ACM Asia Regional (Kolkata Site) Programming Contest December 26, 2015

Instructions

There are **Eleven (11)** problems for each team to be completed in **five hours**. Standard Input and Output files are to be used for each problem. If you test your program using CodeChef platform, it will automatically redirect input from the sample input file to your program. Output must correspond exactly to the provided sample output format, including (mis)spelling and spacing. Multiple spaces will not be used in any of the judges' output, except where explicitly stated.

Your solution to any problem should be submitted for judging using the codechef platform. Once you have submitted the solution, it will reach the codechef judge and the result will be shown. The judgment may be "Correct Answer", meaning that your submission was judged to be correct. Otherwise you will get a message indicating the problem with your program. For example, the message may be "Wrong Answer", "Compilation Error", "Runtime Error", "Time Limit Exceeded" etc.

You can submit only one source file.

You can use any of the standard library functions that your chosen programming language provides. In addition, you can use the math library in C/C++. You cannot use any other library that requires an extra flag to be passed to the compiler command. If you do this, the judges will probably find a code "compilation error" in your program. Your program is not permitted to invoke any external programs. *Violation of this rule may lead to disqualification from the contest.*

Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required. The judges will only test whether the input-output behavior of your program is correct or not. The regional contest director and judges are empowered to adjust for or adjudicate unforeseen events and conditions. Their decisions are final.

Teams are ranked according to the most problems solved. Teams who solve the same number of problems are ranked by least total time. The total time is the sum of the time consumed for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the accepted run plus 20 penalty minutes for every rejected run for that problem regardless of submittal time. There is no time consumed for a problem that is not solved.

Teams are allowed to bring 25 pages of printed materials stamped by the authority in the contest area. Further, the team is not allowed to discuss/ talk with any other team by any means whatsoever, during the contest period. *Any such attempt, if it is detected, may lead to immediate disqualification of all the teams involved.*

In a planet far far away

Problem code: KOL1501

CodeChef has sent Jalebi Bai to a new planet to learn a special recipe from them. On her journey from the space station to the head chef's house, she observed that the multiplication rule on this new planet is slightly different from our Earth.

For example: on this planet, 2*4 may be 9 instead of 8. Weird, she thought.

Curious, Jalebi Bai asked a co-passenger the results of all the 100 multiplications between all pairs of digits.

She figured out that the Standard algorithm for multiplication on this planet follows the same rules as on Earth except that digit multiplication should follow the rule of that planet.

She also found out that any digit multiplied with 0 is 0, the multiplication of x with y is the same as the multiplication of y with x.

Addition is defined to be same as that on the Earth.

For example, if the result of 2 multiplied with 4 is 9, 5 multiplied with 3 is 19, 2 multiplied with 3 is 17 and 4 multiplied with 5 is 24, then multiplication of 25 with 43 will give:

```
          25
          43
-------------------
          19
         170
         240
         900
---------------------
        1329
---------------------
```

The final answer will be 1329 (following the addition algorithm of the earth for managing carry).

Finally Jalebi Bai meets the head chef of the planet, but the problem is he is ready to give her the recipe only if she can solve a puzzle. He asks her to find the sum of the results of multiplication (according to the planet) between all possible ordered pairs of numbers from the set of positive integer less than or equal to a given number **A**. **Note:** If **i** and **j** are distinct, then the pair (**i**,**j**) is considered different from (**j**,**i**).

Jalebi Bai then asks him for a formal multiplication algorithm of the planet, to which the head chef agrees and gives following description.

The multiplication of two numbers **X** and **Y** represented as arrays, with **M** as the digit multiplication matrix, is defined as follows:

```
Multiply(X[1..p], Y[1..q])
{
   m = [1..p+q]  //Allocate space for result
   for b = 1 to q
   {
      carry = 0
      for a = 1 to p
      {
         m[a + b - 1] += carry + M[X[a]] [Y[b]]
         carry = m[a + b - 1] /10
         m[a + b - 1] = m[a + b - 1] mod 10
      }
      m[b + p] += carry
   }
   answer = 0
   for r = p+q to 1
      answer = answer*10 + m[r]
   return answer
}
```

Aliens cannot hear any large number, so they want to hear the output modulo **Mod**, where **Mod** is some integer.

If Jalebi Bai gives a wrong output, she would be eaten up! (Surprise surprise!)

Jalebi Bai can contact someone on earth for help. She reaches you for a code which can evaluate the output for any given input. Can you help her solve this puzzle?

Input

The digit multiplication rule appears in 9 lines. Line **i** contains the multiplication rule for the digit **i**. Each line contains **9** positive integers, each of which is less than **100**. It is guaranteed that the given matrix is symmetric.

The next line contains **T**, the number of test cases to evaluate.

The first line of each test case contains a positive integer **A**.

The second line of each test case contains **Mod**.

Output

For each test case, output the solution modulo **Mod** in a single line.

Constraints

- **1 ≤ T ≤ 50000**
- **1 ≤ A < 10^{100000}**
- **1 ≤ Mod ≤ 10^{8}**
- **1 ≤ Sum of lengths of all strings A ≤ 5*10^{5}**

Example1

Input:

```
1 2 3 4 5 6 7 8 9
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
3
9
10000
73
100
83
100
```

Output:

```
2025
1
96
```

Example2

Input:

```
1 1 1 1 1 1 1 1 1
1 4 6 8 10 12 14 16 18
1 6 9 12 15 18 21 24 27
1 8 12 16 20 24 28 32 36
1 10 15 20 25 30 35 40 45
1 12 18 24 30 36 42 48 54
1 14 21 28 35 42 49 56 63
1 16 24 32 40 48 56 64 72
1 18 27 36 45 54 63 72 81
1
9
10000
```

Output:

```
1953
```

Explanation

Example case 1. Normal multiplication rule follows.

Example case 2. The only pairs with multiplication different from normal multiplication are (1, 2), (1, 3), ..., (1, 9) and (2, 1), (3, 1), ..., (9, 1).

Time limit:	1s
Source limit:	50000B
Languages:	C C++14 JAVA

The Christmas Gift

Problem code: KOL1502

CodeChef has received a package of **N** ingredients from Aloo uncle and Kachori aunty as their Christmas gift. **CodeChef decides to make dishes with every possible combination of these N ingredients**. (**Note:** A dish with 0 ingredients is also possible. CodeChef uses it as an excuse for serving air to their airy customers). Every ingredient from the package has a taste score between **1 and 10**.

Now CodeChef has customers on two planets, planet **A** and planet **B**. People from planet **A** like all the ingredients very much. And hence for every dish given to them, planet **A** will pay CodeChef an amount, which, in Alterian dollars, equals the sum of the taste scores of all the ingredients present in the dish minus the sum of the taste scores of all the ingredients not present in the dish.

People from planet **B** don't like the ingredients at all. So for every dish given to planet **B**, planet **B** will pay CodeChef Alterian dollars equal to the sum of the taste scores of all the ingredients not present in the dish minus the sum of the taste scores of all the ingredients present in the dish.

CodeChef can only make a single dish from a particular combination of ingredients. And **they can send a dish either to planet A or planet B, but not both**. You have to find out the maximum amount of money CodeChef will make by distributing **all the dishes made** with these ingredients on planet **A** and planet **B**.

Report the maximum amount **modulo** $\mathbf{10^7}$.

Input

The first line contains **T**, the number of test cases.

Each test case begins with **N**, the number of ingredients

The next line for the test case contains **N** space-separated integers, which are the taste scores of the ingredients.

Output

For each test case, output the value as asked in a separate line.

Constraints

- **1 ≤ T ≤ 100**
- **1 ≤ N ≤ 1000**
- **1 ≤ Taste scores of ingredients ≤ 10**
- **1 ≤ Sum of N over all Test Cases ≤ 1000**

Example

Input:

1

2

1 2

Output:

8

Explanation

Example case 1. The dishes made by CodeChef and the amounts collected:

- Dish 1: Contains both ingredients: Sold to Planet A for 3 Alterian dollars.
- Dish 2: Contains first ingredients: Sold to Planet B for 1 Alterian dollars.
- Dish 3: Contains second ingredients: Sold to Planet A for 1 Alterian dollars.
- Dish 4: Does not contain any ingredients: Sold to Planet B for 3 Alterian dollars.

Total Amount : 8 Alterian dollars.

Time limit:	1s
Source limit:	50000B
Languages:	C C++14 JAVA

The Parenthesis Tree

Problem code: KOL1503

Samosa Bhai and Jalebi Bai are taking cooking classes from CodeChef. CodeChef, as the name suggests, is also a coding enthusiast. So he decides to get his pupils interested in coding. For this, he takes them both to a strange tree made up of **N** nodes. Each of these nodes is of one of the following two types — nodes containing open parenthesis **‘(’** and nodes containing closed parenthesis **‘)’**. CodeChef asks the students **Q** queries, where in each query they have to find out if the path between two given nodes is a balanced parentheses string or not. If they solve all the queries, they will get to eat the special Christmas cake made by CodeChef. Samosa Bhai and Jalebi Bai are lazy kids, but they also want to eat the cake. So they ask you for help.

Note: A balanced parentheses string means that each opening parenthesis has a corresponding closing parenthesis and the pairs of parentheses are properly nested.

Input

The first line contains **T**, the number of test cases to follow.

Each test case begins with **N** and **Q**, the number of nodes in the tree and the number of queries to follow.

N-1 lines follow. Each line contains 2 space-separated integers, **x** and **y**, which denotes that there is an edge between them.

The next line contains **N** space-separated parentheses. The **ith** parenthesis denotes the value on the **ith** node.

The **Q** queries follow. Each query contains two space-separated integers **u** and **v**.

Output

For each query, output "**Yes**" [without quotes] if the path from **u** to **v** is a balanced parentheses string and "**No**" [without quotes] if the path from **u** to **v** is not a balanced parentheses string.

Constraints

- **1 ≤ T ≤ 1000**
- **1 ≤ N ≤ 10^5**
- **1 ≤ Q ≤ $3*10^5$**
- **1 ≤ Sum of N over all cases ≤ 10^5**
- **1 ≤ Sum of Q over all cases ≤ $3*10^5$**

Example

Input:

```
1
6 4
1 2
2 5
1 3
3 6
3 4
) ( ) ) ( (
5 3
3 5
2 4
6 2
```

Output:

```
Yes
No
No
No
```

Explanation

Example case 1.

- **Query 1:** The path contains (()).
- **Query 2:** The path contains))((.
- **Query 3:** The path contains ())).
- **Query 4:** The path contains ())(.

NOTE: Input files can be large. Please use fast input methods (E.g. scanf in C / C++, BufferedReader in Java)

Time limit:	2s
Source limit:	50000B
Languages:	C C++14 JAVA

The making of a cake

Problem code: KOL1504

CodeChef has received an order from the President of Mithai country for his son's birthday cake. The president is a person of very high temper and CodeChef doesn't want to tick him him, so he had to prepare a cake exactly as described by the President's son. He asked for a cake with **N** layers and each layer has to be of a type specified by him. The type of layer is represented by a lowercase letter from the English alphabet.

CodeChef asked his sous Chef, Jalebi Bai, to make this cake, who was very sleepy due to a very long and tiring journey to a planet far far away earlier. Due to tiredness, Jalebi Bai screwed up the the layers while baking the cake. Thankfully, it has the same number of layers as required, but any of the layers may or may not be the same as described in the order. CodeChef is really worried because of this, as making a new cake will cost him a huge amount of money.

At this point of time, Samosa Bhai comes to the rescue. He has a layer swapper (patent pending) which can swap the layers of a cake without ruining the cake. This swapper has a limitation that it can swap layers separated **exactly** by distance **D** only, meaning there should be **exactly D-1** layers in between the two layers to be swapped.

You have to tell if the cake made by Jalebi Bai can be changed into the cake described by the President's son using Samosa Bhai's swapper.

Input

The first line contains **T**, the number of test cases.

The first line of each test case contains **N** and **D**.

The next line contains a lowercase string **A** representing the cake described by the President's son.

The next line contains a lowercase string **B** representing the cake made by Jalebi Bai.

Output

For each test case, output "**Yes**" [without quotes] if it is possible to transform the cake **B** into cake **A** using the layer swapper. Otherwise, output "**No**" [without quotes].

Constraints

- **$1 \le T \le 20000$**
- **$1 \le N \le 10^5$**
- **$1 \le D \le 10^5$**
- **Size of String A = Size of String B = N**
- **$1 \le$ Sum of N $\le 10^5$**
- **All characters are from 'a' to 'z' in string A and B.**

Example

Input:

```
5
4 2
qnyu
ynqu
4 1
fbnc
nbcf
5 2
abcde
edacb
5 2
abcde
edabc
3 1
eff
bae
```

Output:

```
Yes
Yes
No
Yes
No
```

Explanation

Example case 1. Swap layer 'y' with 'q'.

Example case 2. Follow the following swap order:

- Swap Layer 'f' with 'c', The cake will now be "nbfc"
- Swap Layer 'n' with 'b', The cake will now be "bnfc"
- Swap Layer 'n' with 'f', The cake will now be "bfnc"
- Swap Layer 'f' with 'b', The cake will now be "fbnc"

Example case 4. Follow the following swap order:

- Swap Layer 'b' with 'd', The cake will now be "ebadc"
- Swap Layer 'a' with 'e', The cake will now be "abedc"
- Swap Layer 'e' with 'c', The cake will now be "abcde"

Time limit:	1s
Source limit:	50000B
Languages:	C C++14 JAVA

Jalebi Bai and Decks of Sweets

Problem code: KOL1505

Jalebi Bai has two decks of sweets. These decks are represented by two strings **s** and **t** consisting of lowercase letters from the English alphabet. Each letter from 'a' to 'z' denotes a single type of sweet, e.g., ‘b’ denotes Barfi, 'r' denotes Rasgulla, etc. She wants to use these decks in her friend Laddu’s marriage. Instead of graciously accepting the gift, Laddu asks Jalebi Bai to first make the two decks exactly the same, so that they look good while serving. Now Jalebi Bai can apply the following operation as many times as she wants on both the decks.

- From a single deck, pick any two consecutive sweets of the same type and eat one of them. e.g., if a deck is "rrrjj", she can change it to "rrrj" by picking the last two sweets of type 'j' (Jalebi) and eating one of them, or she can change the deck to "rrjj" by picking up the first two sweets of type ‘r’ and eating one of them.

You need to tell whether Jalebi Bai can make the decks equal by applying the above operation as many times as she wants. Output "Yes" [without quotes] if she can, otherwise output "No" [without quotes].

Input

The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows.

The first line of each test case contains a string **s**.

The second line of each test case contains a string **t**.

Output

For each test case, output "Yes" [without quotes] or "No" [without quotes] depending on the case.

Constraints

- **1** ≤ **T** ≤ **50**
- Each character of **s, t** will be between 'a' and 'z'.
- **1** ≤ length of string **s** ≤ **50**
- **1** ≤ length of string **t** ≤ **50**

Example

Input:

```
2
rrrjj
rrrj
rj
jr
```

Output:

```
Yes
No
```

Explanation

In the first example, Jalebi Bai can change the first deck from "rrrjj" to "rrrj" in a single operation, then both the decks will be equal. Example taken from the statements.
In the second example, Jalebi Bai can never make both decks of sweets equal.

Time limit:	1s
Source limit:	50000B
Languages:	C C++14 JAVA

Samosa Bhai and his Courier Company

Problem code: KOL1506

Samosa Bhai got tired of the difficult tasks given by CodeChef along with meagre salary being paid to him. So he moves to Aloo uncle's village and opens a courier company there. In the village, everyone knows Aloo uncle and Kachori Aunty, so everyone sends their gifts using Samosa Bhai's courier company only.

There are **N** houses in a row in the village. On New Year's Eve, each house sends a gift to all the other houses. The distance between two houses situated at positions **x** and **y** in the row is **|x - y|**. Samosa Bhai's courier company charges $\mathbf{d^k}$ rupees for each delivery, where **d** is the distance between the 2 houses.

Everyone chose Samosa Bhai's company because of Aloo uncle and Kachori aunty. So he decides to give them the amount of money earned by him modulo **1000000007** $\mathbf{(10^9 + 7)}$. You need to compute this amount.

Input

- The first line of input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows.
- The first line of each test case contains 2 integers — **N** and **k** — denoting the number of houses and the exponent **k**, respectively. The second line contains **N** space-separated integers $\mathbf{A_1}$, $\mathbf{A_2}$, … , $\mathbf{A_N}$ denoting the positions of the houses.

Output

- For each test case, output a single line with the amount given to Aloo uncle and Kachori aunty by Samosa Bhai.

Constraints

- $\mathbf{1 \le T \le 10^4}$
- $\mathbf{2 \le N \le 10^5}$
- $\mathbf{0 \le k \le 100}$
- $\mathbf{0 \le A_i \le 1000000006}$
- The sum of **N** over all test cases will be at most **100000**
- 2 houses **can** be at the same position

Example

Input:

```
5
2 2
7 2
2 3
7 2
3 2
1 3 2
10 2
1 2 3 4 5 6 7 8 9 10
10 10
1 2 3 4 5 6 7 8 9 10
```

Output:

```
50
250
12
1650
558199159
```

Explanation

Example case 1. House 1 sends a gift to House 2 with cost $(|2 - 7|)^2$ and house 2 sends a gift to house 1 with cost $(|7-2|)^2$, for a total cost of 50
Example case 3. The cost for gift from house with index 1 to house with index 2 is $2^2 = 4$, from houses 1 to 3 is 1, from houses 2 to 3 is 1. Taking into account the gifts sent the other way (from 2 to 1, 3 to 1 and 3 to 2), the total cost is 4 + 1 + 1 + 4 + 1 + 1 = 12

Time limit:	1.5s
Source limit:	50000B
Languages:	C C++14 JAVA

The Christmas Cookies

Problem code: KOL1507

CodeChef has prepared christmas cookies for everyone in the town. He gives the responsibility of distributing the cookies to Samosa Bhai.

There are **N** houses and **E** roads in the town. Each road starts from one house and ends at another and there are no other houses on the road. **There can be more than one direct road between two houses**. Each road has some road tax. Once you have paid the road tax for a road, you can travel on that road any number of times. The problem is, a Christmas tree has to be installed in the town. It can only be installed on one of the roads resulting in that road being blocked. As a result, Samosa Bhai will not be able to use that road for travelling.

Now, **for each road**, Samosa Bhai wants to know the minimum amount of road tax he will have to pay to distribute candies in all the houses, if the Christmas tree were to be placed on that road. Samosa Bhai can start his journey from any of the houses.

Input

The first line contains **T**, the number of test cases to follow.

The first line of each test case contains the integers **N** and **E** separated by a space.

Next, **E** lines follow. Each line contains three space-separated integers — **x**, **y**, and the road tax of a direct road between house **x** and house **y**.

Output

For each road (in the **same order as input**), output the value of the minimum road tax paid if the christmas tree is placed on that road and it is still possible to distribute cookies in all houses, otherwise output **-1**.

Constraints

- **1 ≤ T ≤ 1000**
- **1 ≤ N ≤ 10^5**
- **1 ≤ E ≤ $2*10^5$**
- **1 ≤ Sum of E over all cases ≤ $2*10^5$**
- **1 ≤ Sum of N over all cases ≤ 10^5**
- **1 ≤ Road Tax ≤ 10^8**

Example

Input:

```
1
5 7
1 2 2
1 4 4
2 3 6
2 5 8
3 5 10
4 3 12
1 5 14
```

Output:

```
30
28
24
22
20
20
20
```

Explanation

Example case 1

- Road from House 1 to House 2 (i.e Road 1) is blocked:
  Use Roads 2, 3, 4 and 6. Total Tax paid = 30

- Road from House 1 to House 4 (i.e Road 2) is blocked:
  Use Roads 1, 3, 4 and 6. Total Tax paid = 28

- Road from House 2 to House 3 (i.e Road 3) is blocked:
  Use Roads 1, 2, 4 and 5. Total Tax paid = 24

Time limit:	1.5s
Source limit:	50000B
Languages:	C C++14 JAVA

Antichains

Problem code: KOL1508

Given an integer **N**, let $\mathbf{F_N}$ be the set of factors of **N**. e.g., F_6 = {1,2,3,6}.
The radical of an integer **N**, denoted by **rad(N)**, is defined as the product of the distinct prime factors of **N**. E.g., rad(12) = 2 * 3 = 6.
Define an **antichain** of a set **S** of integers to be a subset of **S** such that for any two elements x and y in the antichain, rad(x) and rad(y) do not divide each other. e.g., antichains in F_6 are {}, {1}, {2}, {3}, {6} and {2,3}.
Given **N**, find the size of the largest antichain of F_N, and the number of antichains of F_N of that size. Since the answers can be large, print both of them modulo **(10^9+7)**.
e.g., if N=6, the largest antichain is of size 2, and there is only 1 antichain of that size.
Since **N** can be large, the input is the prime factorization of **N**. The input has two arrays of size **M**: **base** and **power**, and N = $base_1^{power\_1}$ * $base_2^{power\_2}$ * ... * $base_M^{power\_M}$

Input

- The first line contains **T**, the number of test cases. Description of the **T** test cases follows.
- Each test case starts with a single integer **M**.
- The next **M** lines each contain 2 integers separated by a space. The i^{th} line contains $\mathbf{base_i}$ and **power_i.**

Output

- For each test case, output one line containing two space-separated integers, respectively the size of the largest antichain modulo (10^9+7) and the number of antichains of that size, again modulo (10^9+7).

Constraints

- **$1 \le T \le 10$**
- **$1 \le M \le 10^5$**
- **$2 \le base_i \le 10^9$**
- **$1 \le power\_i \le 10^9$**
- **$base_i$ is a prime number for all $1 \le i \le M$**
- **For any $1 \le i \ne j \le M$, $base_i \ne base_j$**

NOTE: Input files can be large. Please use fast input methods (E.g. scanf in C / C++, BufferedReader in Java)

Example

Input:

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

Output:

```
2 1
2 2
```

Explanation

Example case 1. The largest antichain size is 2, and there is only one of it {2,3}

Example case 2. The largest antichain size is 2, and there are two of those: {2,3} and {4,3}

Time limit:	1.5s
Source limit:	50000B
Languages:	C C++14 JAVA

Save The Trees

Problem code: KOL1509

CodeChef has been very busy with his christmas preparations and he doesn't have time to look after Samosa Bhai and Jalebi Bai. To keep them busy, CodeChef has given them an array **A** of size **N**. He has asked them to plant trees at the points with Cartesian coordinates (**A[i]**, **A[j]**), such that **i** < **j**.
There are a lot of giraffes nearby. To save the trees from the giraffes, they decide to build a fence around the trees. Moreover, they want to use the minimum length of fencing for this task. Find the value equal to **twice the area** covered by the fence using the minimum length of fencing.

Input

The first line contains **T**, the number of test cases to follow.
The first line of each test case contains an integer **N**, the size of the array.
The second line of the test case contains **N** space-separated integers.

Output

For each test case, output the value equal to **two times the area**, rounded to the nearest integer, surrounded by the fence when using the minimum length of net to surround all the trees.

Constraints

- **$1 \leq T \leq 40000$**
- **$2 \leq N \leq 10^5$**
- **$1 \leq$ Value of the array elements $\leq 10^8$**
- **$1 \leq$ Sum of N over all cases $\leq 2*10^5$**

Example

Input:

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

Output:

```
6
13
```

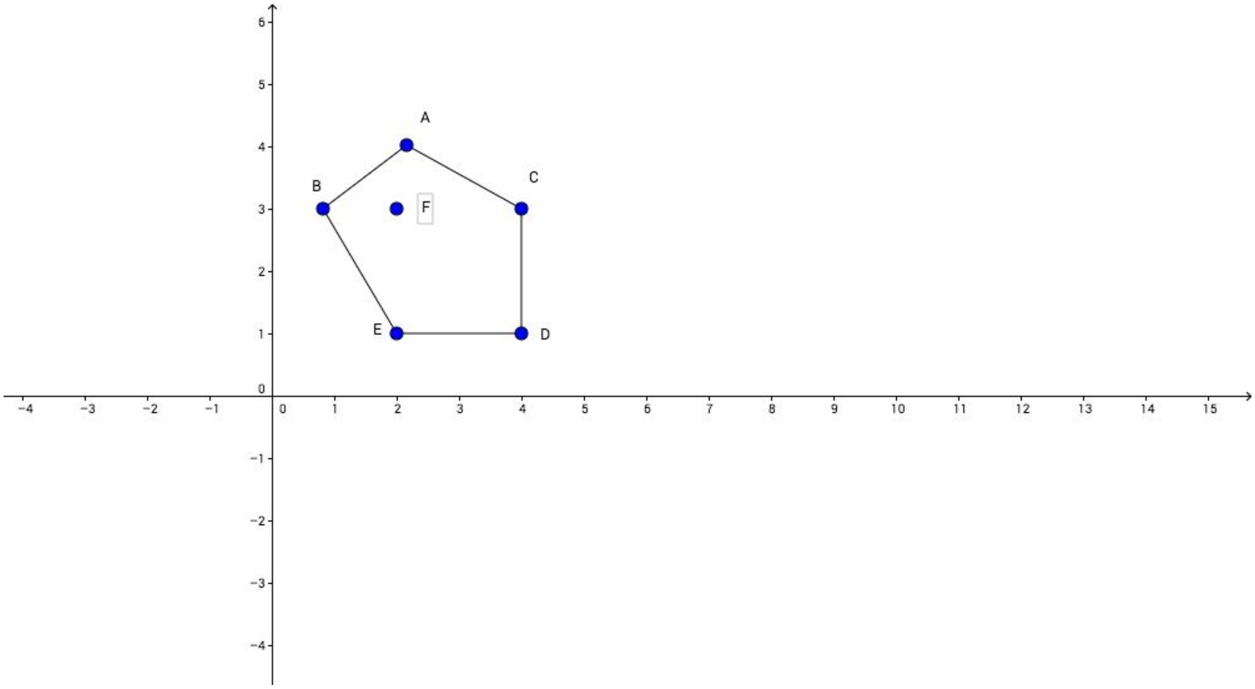
Explanation

Example case 1.

Covered portion is a right angle triangle with vertices (2,4), (2,1) and (4,1).

Area = (1/2)*2*3 = 3

Example case 2.



A, B, C, D, E and F denotes the trees.

The above image denotes the situation. Area of the figure: 6.5

2*(Area of Triangle ABC + Area of Triangle BFE + Area of Square EFCD) is the answer.

Area of Triangle ABC = 1.5

Area of Triangle BFE = 1

Area of Square EFCD = 2*2 = 4

Time limit:	1s
Source limit:	50000B
Languages:	C C++14 JAVA

The Revenge of Jalebi Bai

Problem code: KOL1510

Jalebi Bai recently met her long time friend Barfi Tai. Instead of entertaining her guest, Barfi Tai kept boasting about her **K** marvelous necklaces made of black and golden beads. Jalebi Bai felt really jealous of it and decided to extract revenge.

She has invited Barfi to visit her home next week. Meanwhile, she is planning to buy a truly beautiful necklace to make Barfi jealous. According to her, a truly beautiful necklace should contain each of Barfi's necklaces as a **subsequence** of it. Though still trying to show off, Jalebi Bai is smart and does not want to put a lot of money in it. So she wants to buy a truly beautiful necklace containing the minimum number of beads.

She goes to Devu Sunar and asks him to provide her such a necklace. Devu Sunar is busy this week and has asked you to help him in building the necklace. Please help him!

Note

- A subsequence is a sequence that can be derived from another sequence by deleting some elements but without changing the order of the remaining elements. For example, the sequence [A,B,D] is a subsequence of [A,B,C,D,E,F].
- The bead patterns on the necklaces are **not** considered circular.

Input

The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases is as follows.

The first line of each test case contains an integer **K** denoting the number of necklaces of Barfi. **K** lines follow. Each of them contains a string made from the characters 'B' (representing black bead) and 'G' (representing golden bead), with the **i**th line denoting Barfi Tai's **i**th necklace.

Output

For each test case, output a truly beautiful necklace with the minimum number of beads. If there are more than one such necklace, you are allowed to print any of them.

Constraints

- **1** ≤ **T** ≤ **30**
- **1** ≤ **K** ≤ **16**
- **1** ≤ length of each necklace of Barfi ≤ **8**

Example

Input:

```
3
2
BG
GB
2
BGB
GG
3
BG
GBB
BGB
```

Output:

```
BGB
BGGB
BGBB
```

Explanation

In the first example, Devu can give necklace the BGB to Jalebi Bai, as it contains both the Barfi's necklace BG and GB as subsequence. Note that Devu can also give GBG to Jalebi Bai.
In the second example, Devu can give the necklace BGGB, as it contains both the Barfi's necklace BGB and GG as subsequence. Note that there are many other possible truly beautiful necklaces of length 4, that Devu can give, e.g. BGBG or GBGB. You are allowed to print any of them.
In the third example, Devu can give the necklace BGBB or GBGB. You are print any of them as your answer.

Time limit:	3s
Source limit:	50000B
Languages:	C C++14 JAVA

Wait for it

Problem code: KOL1511

Given **N**, **A** and **B**, find the value of the following expression:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} gcd(A^i - B^i, A^j - B^j)$$

Since the value can be large, find it **modulo (10^9+7).**

Input

- The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows.
- Each test case consists of a single line containing three space-separated integers — **A**, **B**, and **N**.

Output

- For each test case, output a single line containing the value of the expression **modulo 10^9+7**.

Constraints

- **$1 \le T \le 20$**
- **$1 \le N \le 10^9$**
- **$1 \le B < A \le 10^9$**
- **GCD(A,B) = 1, i.e., A and B are coprime.**

Example

Input:

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

Output:

```
8
6
```

Explanation

Example case 1. The summation expands to gcd(1,1) + gcd(5,1) + gcd(1,5) + gcd(5,5) = 8.

Example case 2. The summation expands to gcd(1,1) + gcd(3,1) + gcd(1,3) + gcd(3,3) = 6.

Time limit:	3s
Source limit:	50000B
Languages:	C C++14 JAVA