

## Problem A. A Greeting from Qinhuangdao

Input file:           standard input  
Output file:         standard output  
Time limit:          2 seconds  
Memory limit:       512 mebibytes

*A rainstorm sweeps down on this northern land,  
White breakers leap to the sky.  
No fishing boats off Chinwangtao  
Are seen on the boundless ocean.  
Where are they gone?*

---

Peitaiho To the tune of Lang Tao Sha  
Mao Zedong, Summer 1954

Welcome to the CCPC Qinhuangdao Site!

Qinhuangdao is a beautiful coastal city full of charm, integrating historical heritage and modern civilization. It was named after the Emperor QinShiHuang's east tour in 215 BC for seeking immortals.

The infiltration of more than 2000 years of history has left a rich cultural treasure here. Bo Yi, Shu Qi, Qi Jiguang, Cao Cao, and Mao Zedong, Many Heroes throughout the ages have endowed Qinhuangdao with the thousand-year cultural context, the unique and precious heritage, and the profound historical memory.



Pleasant natural scenery has shaped her beautiful appearance. Thousands of miles of Yan Mountains, the Great Wall, and the vast seas are miraculously met here. The blue sky, green land, blue sea, and golden sand gather together to welcome guests.

To toast your arrival, Alex prepared a simple problem to help you warm up.

Alex has  $r$  red balls and  $b$  blue balls. Then, Alex randomly chose two of these balls with equal probability. What is the probability that he chose two red balls?

Output the required probability in the form of irreducible fraction.

## Input

The first line of input gives the number of test cases,  $T$  ( $1 \leq T \leq 10$ ).  $T$  test cases follow.

For each test case, the only line contains two integers  $r, b$  ( $1 \leq r, b \leq 100$ ), where  $r$  is the number of red balls and  $b$  is the number of blue balls.

## Output

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 answer in the form of irreducible fraction in format  $A/B$ .

If the required probability equals to zero, output  $0/1$ . If the required probability equals to 1, output  $1/1$ .

## Example

standard input	standard output
3	Case #1: 0/1
1 1	Case #2: 1/3
2 1	Case #3: 7/30
8 8	

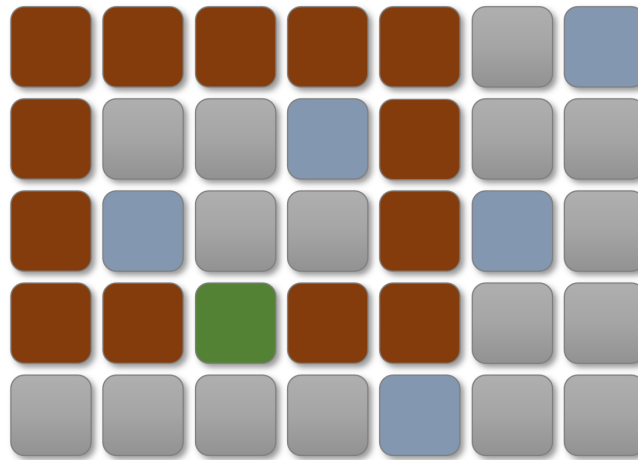
## Problem B. Bounding Wall

Input file: standard input  
Output file: standard output  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

Alex is a professional computer game player.

These days, Alex is playing a war strategy game. His land is a rectangular grid with  $n$  rows by  $m$  columns of cells. He wants to build a bounding wall to protect some vital materials.

The bounding wall is a frame of  $a \times b$  rectangle whose width is 1. It occupies  $a$  rows and  $b$  columns, and its coverage area is  $a \cdot b$ . Note that  $a = 1$  or  $b = 1$  is also allowed. Each cell has a state, wet or dry. It is impossible to build a bounding wall across a wet cell.



He will also have several queries about building a bounding wall. The only two possible formats about queries are listed as follows.

- $1 \ x \ y$  : the state of cell  $(x, y)$  changes.
- $2 \ x \ y$  : Alex wants to build a bounding wall across the cell  $(x, y)$  such that the coverage area is as large as possible. Answer the maximum coverage area.

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 1000$ ).  $T$  test cases follow.

For each test case, the first line contains three integers  $n, m$  and  $Q$  ( $1 \leq n, m, Q \leq 10^3$ ), representing his land has  $n$  rows and  $m$  columns, and he will have  $Q$  queries.

Each of the following  $n$  lines contains a string  $s_{i,1}s_{i,2} \cdots s_{i,m}$  ( $s_{i,j} = \text{'\#'} \text{ or } \text{'.'}$ ), describing the initial state of his land. If  $s_{x,y} = \text{'\#'}$ , the cell  $(x, y)$  will be dry, otherwise, it will be wet.

Each of the following  $Q$  lines contains three integers  $t_i$  ( $1 \leq t_i \leq 2$ ),  $x_i$  ( $1 \leq x_i \leq n$ ) and  $y_i$  ( $1 \leq y_i \leq m$ ), representing a query.

The sum of  $\max\{n, m, Q\}$  in all test cases doesn't exceed  $2 \times 10^4$ .

### Output

For each test case, the output starts with a line containing "Case #x:", where x is the test case number (starting from 1). For each queries with  $t_i = 2$ , answer the maximum possible coverage area. If it is impossible to build a bounding wall, print 0.

## Example

standard input	standard output
2	Case #1:
2 3 2	4
###	3
##.	Case #2:
2 2 2	0
2 1 3	9
4 3 3	
###	
#. #	
#. #	
###	
2 3 2	
1 3 2	
2 3 2	

## Problem C. Cameraman

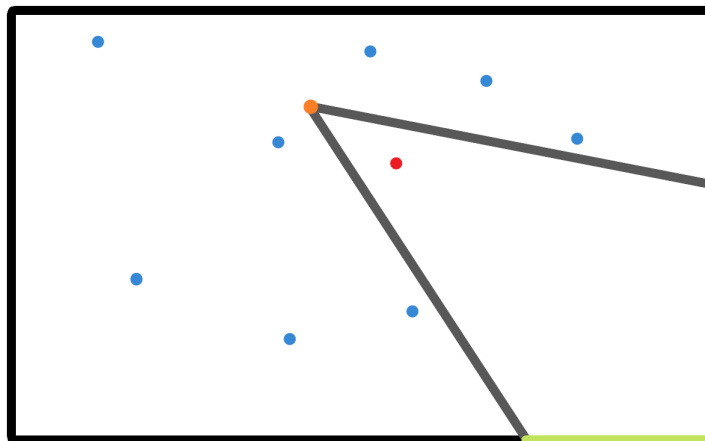
Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Alex is a famous senior cameraman.

His friend, Bob, invites Alex to take a photo of himself. Bob's house can be regarded as a rectangle on the Cartesian plane. Its lower-left corner is  $(0, 0)$ , and its upper-right corner is  $(w, h)$ . Bob is standing on point  $(x, y)$ .

If Alex stands on point  $(u, v)$ , the scope must be an angle of which the vertex is  $(u, v)$ . Since photos can be spliced, the degree of angle can exceed  $180^\circ$ . Bob must be inside this angle (including the boundary).

Bob has  $n$  private objects in his house, and the  $i^{\text{th}}$  object is located at point  $(a_i, b_i)$ . Those private objects must be out of the scope of the camera. Bob and his objects are strictly inside his house, and they are located at different position.



Bob has another strange demand, that the total length of the wall being photographed is as long as possible. Please output the maximum length.

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^4$ ).  $T$  test cases follow.

For each test case, the first line contains two integers  $w, h$  ( $2 \leq w, h \leq 10^4$ ), representing the size of Bob's house.

The second line contains two integers  $x, y$  ( $1 \leq x \leq w - 1, 1 \leq y \leq h - 1$ ), describing Bob's position.

The third line contains an integer  $n$  ( $1 \leq n \leq 10^5$ ), where  $n$  is the number of Bob's private objects.

Each of the following  $n$  lines contains two integers  $a_i, b_i$  ( $1 \leq a_i \leq w - 1, 1 \leq b_i \leq h - 1$ ), describing the position of the  $i^{\text{th}}$  object.

The sum of  $n$  in all test cases doesn't exceed  $5 \times 10^5$ .

### Output

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 length.

Your answers should have absolute or relative errors of at most  $10^{-6}$ .

## Example

standard input	standard output
2	Case #1: 13.50000000
8 8	Case #2: 16.00000000
3 7	
3	
5 6	
2 7	
2 5	
6 6	
3 2	
5	
5 4	
2 4	
5 5	
2 3	
1 4	

## Problem D. Defend City

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Alex is a professional computer game player.

These days, Alex is playing a war strategy game. His city can be regarded as a rectangle on the Cartesian plane. Its lower-left corner is  $(0, 0)$ , and its upper-right corner is  $(n + 1, n + 1)$ .

Alex builds  $n$  defensive towers to protect the city. The defensive tower  $i$  is located at  $(x_i, y_i)$ , and its direction is  $d_i$ . The defensive towers with different directions protect different areas:

- if  $d_i = 1$ , the defensive tower  $i$  can protect area  $\{(a, b) \mid a \geq x_i, b \geq y_i\}$ ;
- if  $d_i = 2$ , the defensive tower  $i$  can protect area  $\{(a, b) \mid a \leq x_i, b \geq y_i\}$ ;
- if  $d_i = 3$ , the defensive tower  $i$  can protect area  $\{(a, b) \mid a \leq x_i, b \leq y_i\}$ ;
- if  $d_i = 4$ , the defensive tower  $i$  can protect area  $\{(a, b) \mid a \geq x_i, b \leq y_i\}$ .

If Alex launches  $e$  defensive towers, he will consume  $e$  energy per hour. He wants to launch as few defensive towers as possible so that all points  $(x, y)$  ( $x, y \in \mathbb{R}, 0 \leq x, y \leq n + 1$ ) in his city can be protected. Can you find the optimal strategy?

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^4$ ).  $T$  test cases follow.

For each test case, the first line contains an integer  $n$  ( $1 \leq n \leq 10^6$ ), where  $n$  is the number of defensive towers.

Each of the following  $n$  lines contains three integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n$ ) and  $d_i$  ( $1 \leq d_i \leq 4$ ), representing the position and direction of defensive tower  $i$ .

The sum of  $n$  in all test cases doesn't exceed  $5 \times 10^6$ .

### Output

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 minimum number of defensive towers. If it is impossible to protect all the city, output "Impossible" (without quote).

### Example

standard input	standard output
2	Case #1: Impossible
3	Case #2: 4
1 1 1	
2 2 2	
3 3 3	
4	
1 1 1	
2 2 3	
2 1 2	
1 2 4	

## Problem E. Exam Results

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Professor Alex is preparing an exam for his students now.

There will be  $n$  students participating in this exam. If student  $i$  has a good mindset, he/she will perform well and get  $a_i$  points. Otherwise, he/she will get  $b_i$  points. So it is impossible to predict the exam results. Assume the highest score of these students is  $x$  points. The students with a score of no less than  $x \cdot p\%$  will pass the exam.

Alex needs to know what is the maximum number of students who can pass the exam in all situations. Can you answer his question?

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 5 \times 10^3$ ).  $T$  test cases follow.

For each test case, the first line contains two integers  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) and  $p$  ( $1 \leq p \leq 100$ ), where  $n$  is the number of students and  $p\%$  is the ratio.

Each of the following  $n$  lines contains two integers  $a_i, b_i$  ( $1 \leq b_i \leq a_i \leq 10^9$ ), representing the scores of student  $i$ .

The sum of  $n$  in all test cases doesn't exceed  $5 \times 10^5$ .

### Output

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 number of students.

### Example

standard input	standard output
2	Case #1: 2
2 50	Case #2: 4
2 1	
5 1	
5 60	
8 5	
9 3	
14 2	
10 8	
7 6	



## Problem F. Friendly Group

Input file:            standard input  
Output file:           standard output  
Time limit:            1 seconds  
Memory limit:         512 mebibytes

Professor Alex will organize students to attend an academic conference.

Alex has  $n$  excellent students, and he decides to select some of them (possibly none) to attend the conference. They form a group. Some pairs of them are friends.

The friendly value of the group is initially 0. For each couple of friends  $(x, y)$ , if both of them attend the conference, the friendly value of the group will increase 1, and if only one of them attends the conference, the friendly value of the group will decrease 1. If  $k$  students attend the conference, the friendly value of the group will decrease  $k$ .

Alex wants to make the group more friendly. Please output the maximum friendly value of the group.

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^4$ ).  $T$  test cases follow.

For each test case, the first line contains two integers  $n$  ( $1 \leq n \leq 3 \times 10^5$ ) and  $m$  ( $1 \leq m \leq 10^6$ ), where  $n$  is the number of students and  $m$  is the number of couples of friends.

Each of the following  $m$  lines contains two integers  $x_i, y_i$  ( $1 \leq x_i, y_i \leq n, x_i \neq y_i$ ), representing student  $x_i$  and student  $y_i$  are friends. It guaranteed that unordered pairs  $(x_i, y_i)$  are distinct.

The sum of  $n$  in all test cases doesn't exceed  $10^6$ , and the sum of  $m$  in all test cases doesn't exceed  $2 \times 10^6$ .

### Output

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 friendly value of the group.

### Example

standard input	standard output
2	Case #1: 1
4 5	Case #2: 0
1 2	
1 3	
1 4	
2 3	
3 4	
2 1	
1 2	

## Problem G. Good Number

Input file:            standard input  
Output file:          standard output  
Time limit:           2 seconds  
Memory limit:        512 mebibytes

Alex loves numbers.

Alex thinks that a positive integer  $x$  is good if and only if  $\lfloor \sqrt[k]{x} \rfloor$  divides  $x$ .

Can you tell him the number of good positive integers which do not exceed  $n$  ?

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10$ ).  $T$  test cases follow.

For each test case, the only line contains two integers  $n$  and  $k$  ( $1 \leq n, k \leq 10^9$ ).

### Output

For each test cases, output one line containing “Case #x: y”, where x is the test case number (starting from 1), and y is the answer.

### Example

standard input	standard output
2	Case #1: 233
233 1	Case #2: 43
233 2	

## Problem H. Holy Sequence

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Alex is proud of the holy sequence.

Alex thinks that a sequence  $a_1, a_2, \dots, a_n$  is  $n$ -holy if and only if

- For each  $i \in [1, n]$ ,  $a_i \in \{1, 2, \dots, n\}$  ;
- For each  $i \in [1, n]$ ,  $p_i - p_{i-1} \leq 1$  ( $p_k = \max\{a_1, a_2, \dots, a_k\}$ ,  $p_0 = 0$ ).

For each  $t = 1, 2, \dots, n$ , Alex wants to know

$$\sum_{p \in S_n} \text{cnt}(p, t)^2,$$

where  $S_n$  is the set of all  $n$ -holy sequences and  $\text{cnt}(p, t)$  is the occurrence number of  $t$  in sequence  $p$ .

As the results can be very large, output it modulo  $m$ .

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10$ ).  $T$  test cases follow.

For each test case, the only line contains two integers  $n$  ( $1 \leq n \leq 3000$ ) and  $m$  ( $1 \leq m \leq 10^9$ ), where  $n$  is the length of holy sequences and  $m$  is the modulus.

The sum of  $n$  in all test cases doesn't exceed  $10^4$ .

### Output

For each test case, the output starts with a line containing "Case #x:", where  $x$  is the test case number (starting from 1). The second line contains  $n$  numbers, the answer for each  $t = 1, 2, \dots, n$ , modulo  $m$ .

### Example

standard input	standard output
2	Case #1:
3 998	19 7 1
10 998	Case #2:
	996 713 406 353 206 275 175 936 49 1

## Problem I. Interstellar Hunter

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Alex is a professional computer game player.

These days, Alex is playing an interstellar game. This game is played in an infinite 2D universe, and Alex's spaceship starts at  $(0, 0)$  in the beginning. Because the distances between different galaxies are too long (conventional thrusters cannot reach the destination in an acceptable amount of time), movement can only be done by "jump". If his spaceship has jump skill  $(a, b)$  and he is located at  $(x, y)$ , he can reach  $(x + a, y + b)$  or  $(x - a, y - b)$  immediately. He can use jump skills continuously at any time, and the jump skills will not disappear.

In the game, the following  $Q$  events, which have two types, will happen in order.

- 1  $x_i y_i$ : Alex will acquire jump skill  $(x_i, y_i)$ .
- 2  $x_i y_i w_i$ : There will be a reward mission in  $(x_i, y_i)$ . Alex can decide to go there immediately for  $w_i$  rewards or just ignore it.

Alex wonders how many rewards he can get if he plays optimally.

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^4$ ).  $T$  test cases follow.

For each test case, the first line contains a integer  $Q$  ( $1 \leq Q \leq 10^5$ ), where  $Q$  is the number of the following events.

Each of the following  $Q$  lines contains three or four integers  $t_i$  ( $1 \leq t_i \leq 2$ ),  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 10^6$ ) and maybe  $w_i$  ( $1 \leq w_i \leq 10^9$ ), describing the following events.

The sum of  $Q$  in all test cases doesn't exceed  $10^6$ .

### Output

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 rewards Alex can get.

### Example

standard input	standard output
2	Case #1: 2
4	Case #2: 3
1 1 1	
2 3 1 1	
1 1 3	
2 3 1 2	
3	
1 1 1	
1 2 1	
2 3 2 3	

## Problem J. Jewel Splitting

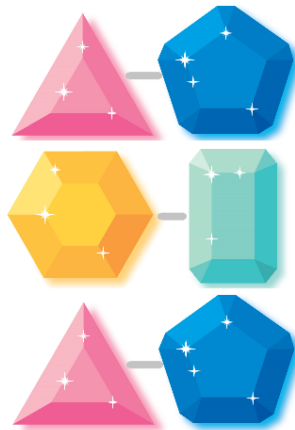
Input file: standard input  
Output file: standard output  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

Alex is a private jewel collector.

Recently, he purchased a piece of jewelry, a bunch of  $n$  gemstones. The gemstones have 26 different types, labeled from **a** to **z**. He wants to split the jewelry and combine them into a jewel matrix.



Alex will choose  $d \in \{1, 2, \dots, n\}$ , the width of the jewel matrix. Then, he will split the jewelry and get  $\lfloor \frac{n}{d} \rfloor$  bunches of gemstones with length  $d$  and a bunch of length  $n \bmod d$  if  $d$  doesn't divide  $n$ . All  $\lfloor \frac{n}{d} \rfloor$  bunches with length  $d$  will be permuted as rows of the jewel matrix.



Alex wonders how many different jewel matrix he can get. Two jewel matrix are considered different if and only if they have different width, or it is at least one row different.

As the results can be very large, output it modulo 998244353.

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 1000$ ).  $T$  test cases follow.

For each test case, the only line contains a string  $s_1s_2 \dots s_n$  ( $1 \leq n \leq 3 \times 10^5, s_i \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots, \mathbf{z}\}$ ), representing the jewelry.

The sum of  $n$  in all test cases doesn't exceed  $10^6$ .

### Output

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 answer modulo 998244353.

## Example

standard input	standard output
2 ababccd aab	Case #1: 661 Case #2: 6

## Problem K. Kingdom's Power

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         512 mebibytes

Alex is a professional computer game player.

These days, Alex is playing a war strategy game. The kingdoms in the world form a rooted tree. Alex's kingdom 1 is the root of the tree. With his great diplomacy and leadership, his kingdom becomes the greatest empire in the world. Now, it's time to conquer the world!

Alex has almost infinite armies, and all of them are located at 1 initially. Every week, he can command one of his armies to move one step to an adjacent kingdom. If an army reaches a kingdom, that kingdom will be captured by Alex instantly.

Alex would like to capture all the kingdoms as soon as possible. Can you help him?

### Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^5$ ).  $T$  test cases follow.

For each test case, the first line contains an integer  $n$  ( $1 \leq n \leq 10^6$ ), where  $n$  is the number of kingdoms in the world.

The next line contains  $(n - 1)$  integers  $f_2, f_3, \dots, f_n$  ( $1 \leq f_i < i$ ), representing there is a road between  $f_i$  and  $i$ .

The sum of  $n$  in all test cases doesn't exceed  $5 \times 10^6$ .

### Output

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 minimum time to conquer the world.

### Example

standard input	standard output
2	Case #1: 2
3	Case #2: 6
1 1	
6	
1 2 3 4 4	

## Problem L. Lost Temple

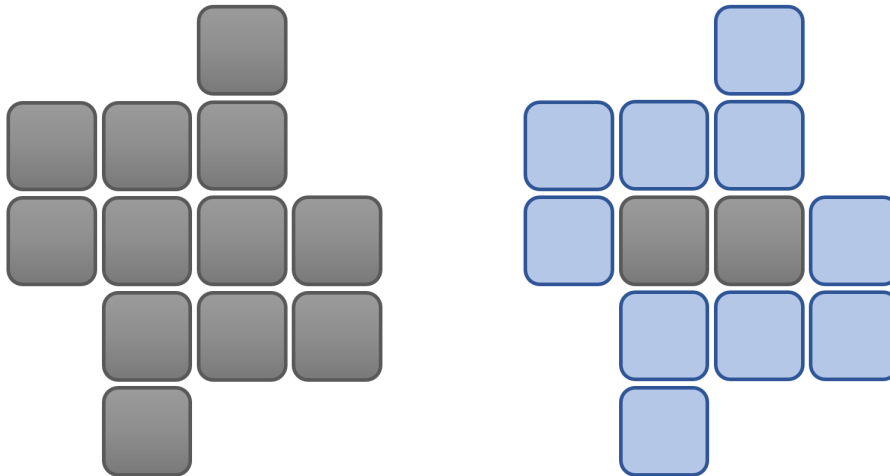
Input file: standard input  
Output file: standard output  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

Professor Alex is an expert in the study of ancient buildings.

When he was surveying an ancient temple, he found ruins containing several fallen stone pillars. There are  $n$  stone pillars in the ruins, numbered  $1, 2, \dots, n$ . For each  $i \in \{1, 2, \dots, n-1\}$ , the  $(i+1)^{\text{th}}$  stone pillar is next to the  $i^{\text{th}}$  stone pillar from west to east. As time went on, some stone pillars were badly destroyed.

Each stone pillar can be regarded as a series of continuous cubic stones, and we label the cubic stones  $1, 2, 3, \dots, 10^9$  from south to north. Every two east-west adjacent cubic stones have the same label. The  $i^{\text{th}}$  stone pillar remains cubic stones  $l_i, l_i + 1, \dots, r_i$ .

Here is an example:  $n = 4, \{l_i\} = \{4, 2, 3, 3\}, \{r_i\} = \{5, 5, 6, 4\}$ ,



Whenever a year goes by, the outer cubic stone will be destroyed by acid rain. In the above example, those blue squares will disappear one year later. Alex wants to know when the all of cubic stones of the  $i^{\text{th}}$  stone pillar will disappear for each  $i = 1, 2, \dots, n$ . Assume that the  $i^{\text{th}}$  stone pillar will disappear  $f_i$  years later. You need to output the answer:

$$\text{answer} = \sum_{i=1}^n f_i \cdot 3^{i-1} \bmod 998244353$$

Since the number of stone pillars can be very large, we generate the test data in a special way. We will provide five parameters  $L, X, Y, A, B$ , and please refer to the code (written in C++) below:

```
1 typedef unsigned long long u64;  
2  
3 u64 xorshift128p(u64 &A, u64 &B) {  
4     u64 T = A, S = B;  
5     A = S;  
6     T ^= T << 23;  
7     T ^= T >> 17;  
8     T ^= S ^ (S >> 26);  
9     B = T;  
10    return T + S;  
11 }
```



```
12 |
13 | void gen(int n, int L, int X, int Y, u64 A, u64 B, int l[], int r[]) {
14 |     for (int i = 1; i <= n; i++) {
15 |         l[i] = xorshift128p(A, B) % L + X;
16 |         r[i] = xorshift128p(A, B) % L + Y;
17 |         if (l[i] > r[i]) swap(l[i], r[i]);
18 |     }
19 | }
```

## Input

The first line of the input gives the number of test cases,  $T$  ( $1 \leq T \leq 10^4$ ).  $T$  test cases follow.

For each test case, the first line contains two integers  $n$  ( $1 \leq n \leq 5 \times 10^6$ ) and  $L$  ( $1 \leq L \leq 5 \times 10^8$ ), where  $n$  is the number of stone pillars and  $L$  is the number of possible value of  $l_i$  and  $r_i$ .

The second line contains two integers  $X$  and  $Y$  ( $1 \leq X, Y \leq 5 \times 10^8$ ), where  $X$  is the minimum possible value of  $l_i$  and  $Y$  is the minimum possible value of  $r_i$ .

The last line contains two integers  $A$  and  $B$  ( $0 \leq A, B < 2^{64}$ ), representing the initial seed.

The sum of  $n$  in all test cases doesn't exceed  $1.2 \times 10^7$ .

## Output

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 answer modulo 998244353.

## Example

standard input	standard output
2	Case #1: 52
4 6	Case #2: 798322
1 1	
43 123	
13 10	
6 1	
4 5	