ArcSoft's Office Rearrangement

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 2848 Accepted Submission(s): 939

Problem Description

ArcSoft, Inc. is a leading global professional computer photography and computer vision technology company.

There are N working blocks in ArcSoft company, which form a straight line. The CEO of ArcSoft thinks that every block should have equal number of employees, so he wants to re-arrange the current blocks into K new blocks by the following two operations:

- merge two neighbor blocks into a new block, and the new block's size is the sum of two old blocks'.
- split one block into two new blocks, and you can assign the size of each block, but the sum should be equal to the old block.

Now the CEO wants to know the **minimum** operations to re-arrange current blocks into K block with equal size, please help him.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with one line which two integers N and K, which is the number of old blocks and new blocks.

The second line contains N numbers $a_1, a_2, \cdots, a_N$, indicating the size of current blocks.

Limits
$1 \le T \le 100$
$1 \le N \le 10^5$
$1 \le K \le 10^5$
$1 \le a_i \le 10^5$

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the minimum operations.

If the CEO can't re-arrange K new blocks with equal size, **y** equals **-1**.

Sample Input

```
3
1 3
14
3 1
2 3 4
3 6
1 2 3
```

Sample Output

```
Case #1: -1
Case #2: 2
Case #3: 3
```

Source

2016年中国大学生程序设计竞赛（杭州）

Bomb

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 2476 Accepted Submission(s): 783

Problem Description

There are N bombs needing exploding.

Each bomb has three attributes: exploding radius r_i, position (x_i, y_i) and lighting-cost c_i which means you need to pay c_i cost making it explode.

If a un-lighting bomb is in or on the border the exploding area of another exploding one, the un-lighting bomb also will explode.

Now you know the attributes of all bombs, please use the **minimum** cost to explode all bombs.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with an integers N, which indicates the numbers of bombs.

In the following N lines, the ith line contains four intergers x_i, y_i, r_i and c_i, indicating the coordinate of ith bomb is (x_i, y_i), exploding radius is r_i and lighting-cost is c_i.

Limits
- $1 \leq T \leq 20$
- $1 \leq N \leq 1000$
- $-10^8 \leq x_i, y_i, r_i \leq 10^8$
- $1 \leq c_i \leq 10^4$

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the minimum cost.

Sample Input

```
1
5
0 0 1 5
1 1 1 6
0 1 1 7
3 0 2 10
5 0 1 4
```

Sample Output

```
Case #1: 15
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Car

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 3074 Accepted Submission(s): 827

Problem Description

Ruins is driving a car to participating in a programming contest. As on a very tight schedule, he will drive the car without any slow down, so the speed of the car is non-decrease real number.

Of course, his speeding caught the attention of the traffic police. Police record N positions of Ruins without time mark, the only thing they know is every position is recorded at an integer time point and Ruins started at 0.

Now they want to know the **minimum** time that Ruins used to pass the last position.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with an integers N, which is the number of the recorded positions.

The second line contains N numbers $a_1, a_2, \cdots, a_N$, indicating the recorded positions.

Limits
$1 \leq T \leq 100$
$1 \leq N \leq 10^5$
$0 < ai \leq 10^9$
$a_i < a_{i+1}$

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the minimum time.

Sample Input

```
1
3
6 11 21
```

Sample Output

```
Case #1: 4
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Difference

Time Limit: 6000/3000 MS (Java/Others) Memory Limit: 65536/65536 K (Java/Others)
Total Submission(s): 1306 Accepted Submission(s): 357

Problem Description

Little Ruins is playing a number game, first he chooses two positive integers y and K and calculates $f(y,K)$, here

$$f(y,K)=\sum_{z \text{ in every digits of } y} z^K (f(233,2)=2^2+3^2+3^2=22)$$

then he gets the result

$$x=f(y,K)-y$$

As Ruins is forgetful, a few seconds later, he only remembers K, x and forgets y. please help him find how many y satisfy $x=f(y,K)-y$.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case contains one line with two integers x, K.

Limits
$1 \le T \le 100$
$0 \le x \le 10^9$
$1 \le K \le 9$

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Sample Input

```
2
2 2
3 2
```

Sample Output

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

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

Equation

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 740 Accepted Submission(s): 221

Problem Description

Little Ruins is a studious boy, recently he learned addition operation! He was rewarded some number bricks of 1 to 9 and infinity bricks of addition mark '+' and equal mark '='.

Now little Ruins is puzzled by those bricks because he wants to put those bricks into as many different addition equations form $x + y = z$ as possible. Each brick can be used at most once and x, y, z are one digit integer.

As Ruins is a beginer of addition operation, x, y and z will be single digit number.

Two addition equations are different if any number of x, y and z is different.

Please help little Ruins to calculate the **maximum** number of different addition equations.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case contains one line with nine integers, the i^{th} integer indicates the number of bricks of i.

Limits
$1 \leq T \leq 30$
$0 \leq$ bricks number of each type ≤ 100

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Sample Input

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

Sample Output

```
Case #1: 2
Case #2: 6
Case #3: 2
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Four Operations

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 2719 Accepted Submission(s): 796

Problem Description

Little Ruins is a studious boy, recently he learned the four operations!

Now he want to use four operations to generate a number, he takes a string which only contains digits **'1'** - **'9'**, and split it into 5 intervals and add the four operations **'+'**, **'-'**, **'*'** and **'/'** in order, then calculate the result(/ used as integer division).

Now please help him to get the largest result.

Input

First line contains an integer T, which indicates the number of test cases.

Every test contains one line with a string only contains digits **'1'-'9'**.

Limits
$1 \leq T \leq 10^5$
$5 \leq \text{length of string} \leq 20$

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Sample Input

```
1
12345
```

Sample Output

```
Case #1: 1
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Game of Eliminate

Time Limit: 10000/5000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 265 Accepted Submission(s): 24

Problem Description

Little Ruins is a studious boy, but in rest time, he will play some little game.

Today he found a game of eliminate: there is $N \times M$ tiles which only contains **'#'** and **'*'**, you have two patterns to eliminate tiles:

:.

and

.:

Each step you can use a pattern and eliminate tiles **on the bottom two lines**. After each step,the tiles above eliminated tiles will fall down.

Your goal is to eliminate all **'*'** tiles, please calculate the **minimum** steps.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with two integers N and M, which indicates the size of tiles.

In the following N lines, every line contains M characters means the type of tiles.

Limits
$1 \leq T \leq 50$.
$1 \leq N \leq 2000$
$2 \leq M \leq 2000$.
For 80% of the use cases, $1 \leq N, M \leq 100$ holds.

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Sample Input

```
1
3 2
#*
*#
##
```

Sample Output

```
Case #1: 2
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

Hazy String

Time Limit: 6000/3000 MS (Java/Others) Memory Limit: 65536/65536 K (Java/Others)
Total Submission(s): 37 Accepted Submission(s): 1

Problem Description

Archaeologists find an ancient string on a monument, but some characters have become hazy with the passage of time, and we only know the remain characters.

Now archaeologists want to re-build the string, and know two rules:

- there is no palindrome substring in the original string. **Note that a single character is not regarded as a palindrome string here**.
- the character set size of the original string is K.
- $0 \leq$ the character of the original string $< K$

Please calculate the number of original string which satisfy above rules.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with three integers N, K and L, which is the number of known characters, the character set size of the original string and the length of the original string.

Then N lines follow, the i^{th} line contains two integers p_i and v_i, means the position and value of i^{th} known character.

Limits
$1 \leq T \leq 100$.
$0 \leq N \leq 2000$.
$1 \leq K \leq 10^9$.
$max(1, N) \leq L \leq 10^9$.
$0 \leq p_i < p_{i+1} < L$.
$0 \leq v_i < K$
For 90\% of the use cases, $N \leq 10$ holds.

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Because **y** could be very large, just mod it with $10^9 + 7$.

Sample Input

```
3
0 3 4
1 4 4
1 1
2 5 5
1 1
3 2
```

Sample Output

```
Case #1: 6
Case #2: 12
Case #3: 27
```

欢迎杭电广大新生参加

Inequality

Time Limit: 4000/2000 MS (Java/Others) Memory Limit: 262144/131072 K (Java/Others)
Total Submission(s): 187 Accepted Submission(s): 17

Problem Description

Little Ruins is a studious boy, recently he learned inequation!

As homework, his teacher gives him a problem of inequation: give you an array a with length $N-1$, please find an array x with length N and $x_i \times x_{i+1} \geq a_i$ for each i from 1 to $N-1$ and try to **minimize** the sum of x.

Input

First line contains an integer T, which indicates the number of test cases.

Every test case begins with an integers N, which is the length of array x.

The second line contains $N-1$ integers $a_1, a_2, \cdots, a_{N-1}$, indicating the array a.

Limits
$1 \leq T \leq 50$.
$1 \leq N \leq 2000$.
$0 < a_i \leq 10^4$.
For 80\% of the use cases, $1 \leq N \leq 100$ holds.

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Round the **y** to the fifth digit after the decimal point.

Sample Input

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

Sample Output

```
Case #1: 5.77350
Case #2: 5.47723
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Just a Math Problem

Time Limit: 20000/15000 MS (Java/Others) Memory Limit: 262144/262144 K (Java/Others)
Total Submission(s): 548 Accepted Submission(s): 175

Problem Description

Little Ruins is a studious boy, recently he learned math!

Now he defines $f(k)$ equal the number of prime factors in k, and $g(k) = 2^{f(k)}$, he want to know

$$\sum_{i=1}^{n} g(i)$$

Please help him!

Input

First line contains an integer T, which indicates the number of test cases.

Every test case contains one line with one integer n.

Limits
$1 \leq T \leq 50$.
$1 \leq n \leq 10^{12}$.

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result.

Because **y** could be very large, just mod it with $10^9 + 7$.

Sample Input

```
3
1
10
100
```

Sample Output

```
Case #1: 1
Case #2: 23
Case #3: 359
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding

Kingdom of Obsession

Time Limit: 2000/1000 MS (Java/Others) Memory Limit: 65536/32768 K (Java/Others)
Total Submission(s): 2092 Accepted Submission(s): 613

Problem Description

There is a kindom of obsession, so people in this kingdom do things very strictly.

They name themselves in integer, and there are n people with their id continuous $(s+1, s+2, \cdots, s+n)$ standing in a line in arbitrary order, be more obsessively, people with id x wants to stand at y^{th} position which satisfy

$$x \mod y = 0$$

Is there any way to satisfy everyone's requirement?

Input

First line contains an integer T, which indicates the number of test cases.

Every test case contains one line with two integers n, s.

Limits
$1 \leq T \leq 100$.
$1 \leq n \leq 10^9$.
$0 \leq s \leq 10^9$.

Output

For every test case, you should output **'Case #x: y'**, where **x** indicates the case number and counts from **1** and **y** is the result string.

If there is any way to satisfy everyone's requirement, **y** equals **'Yes'**, otherwise **y** equals **'No'**.

Sample Input

```
2
5 14
4 11
```

Sample Output

```
Case #1: No
Case #2: Yes
```

Source

2016年中国大学生程序设计竞赛（杭州）

Recommend

liuyiding