

## Problem A. Chat Group

Input file:        standard input  
Output file:       standard output

It is said that a dormitory with 6 persons has 7 chat groups  $\wedge$ . But the number can be even larger: since every 3 or more persons could make a chat group, there can be 42 different chat groups.

Given  $N$  persons in a dormitory, and every  $K$  or more persons could make a chat group, how many different chat groups could there be?

### Input

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

Each test case contains one line with two integers  $N$  and  $K$  indicating the number of persons in a dormitory and the minimum number of persons that could make a chat group.

### 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 number of different chat groups modulo 1000000007.

### Limits

- $1 \leq T \leq 100$ .
- $1 \leq N \leq 10^9$ .
- $3 \leq K \leq 10^5$ .

### Example

standard input	standard output
1 6 3	Case #1: 42

## Problem B. Scapegoat

Input file:        standard input  
Output file:       standard output

Aori is very careless so she is always making troubles. One day she does it again, with  $N$  big troubles! But this time she seems to be at ease because she has found  $M$  Inklings to take all the blames. Each trouble can be measured by a severity number  $a_i$ . Each trouble needs at least one Inking to take the blame, and each Inking can help Aori to take the blame for exactly one trouble. If two or more Inklings take the same trouble, they will take this blame together and discuss how to divide this trouble into.. some trivial things.. to reduce the pressure on each Inking, as long as the total severity on Inklings is equal to the severity of this trouble.

Inklings are so warm so that Aori wants to think a way to let the variance of severity on each Inking to be minimal. Could you help Aori make her scapegoats?

Formally, the variance of variables is the expectation of the squared deviation of a variable from its mean:

$$\sigma^2 = \frac{1}{n} \sum (x_i - \frac{1}{n} \sum x_j)^2$$

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

For each test case, the first line contains two integers  $N$  and  $M$ , where  $N$  is the number of troubles, and  $M$  is the number of Inklings. The next line contains  $N$  integers  $a_1, a_2, \dots, a_N$  representing the severity of the troubles that Aori makes.

### 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 minimal variance.

y will be considered correct if it is within an absolute or relative error of  $10^{-8}$  of the correct answer.

### Limits

- $1 \leq T \leq 100$ .
- $1 \leq N \leq M \leq 2 \times 10^5$ .
- $1 \leq a_i \leq 10000$ .
- $\sum M \leq 3 \times 10^6$ .

### Example

standard input	standard output
3	Case #1: 0.000000000000
3 6	Case #2: 0.400000000000
1 2 3	Case #3: 2.888888888889
5 10	
5 6 7 8 9	
6 6	
1 1 4 5 1 4	

### Note

For the first sample, Aori can let one Inking to take the first trouble’s blame, two Inklings for the second, and three Inklings for the third. The severity on each Inking is 1 unit, so their variance should be 0.

## Problem C. Traffic Light

Input file:        standard input  
Output file:       standard output

One year ago, Mr. Ang joined a great company and he rented a house on the same street as the company. The company is so great that Mr. Ang doesn't need to punch in and out. He can have a good sleep and gets up at any time.

Every day, he walks along the street, goes through  $N$  traffic lights numbered  $1, 2, 3, \dots, N$  and arrives the company. It takes Mr. Ang  $S_0$  seconds from the house to first traffic light.  $S_i$  seconds from traffic light  $i$  to traffic light  $i + 1$  and  $S_N$  seconds from  $N^{\text{th}}$  traffic light to company. The time spent on the way to company depends on the state of traffic lights.

Mr. Ang got interested in the traffic lights and hacked into the system. After reading the code, he found that for traffic light  $i$  on his way, it stays  $A_i$  seconds green and then stays  $B_i$  seconds red and repeats. He also found that for all traffic lights,  $A_i + B_i$  are the same. He can modify the code to set different offsets  $OFF_i$  for the traffic lights. Formally, Mr. Ang can set the offsets for the traffic lights to  $OFF_i$  so that for each traffic light, Mr. Ang can go through it in time range  $[OFF_i + k \times (A_i + B_i), OFF_i + k \times (A_i + B_i) + A_i)$  and must wait in time range  $[OFF_i + k \times (A_i + B_i) + A_i, OFF_i + (k + 1) \times (A_i + B_i))$  for all integers  $k$ .

Now Mr. Ang would like to make his commuting time from house to company as short as possible in the worst case. Find out the minimal time in second.

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each test case starts with a line which consists of one number  $N$ , indicating the number of traffic lights.

The next line consists of  $N + 1$  numbers  $S_0, S_1, \dots, S_N$  indicating the walking time between house, traffic lights and company in seconds as described in the problem.

Then followed by  $N$  lines each consists of 2 numbers  $A_i, B_i$  indicating the length of green light and red light of the  $i^{\text{th}}$  traffic light.

### 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 minimal time in second from house to company in the worst case.

y will be considered correct if it is within an absolute or relative error of  $10^{-8}$  of the correct answer.

### Limits

- $1 \leq T \leq 50$ .
- $1 \leq N \leq 1000$ .
- $1 \leq A_i, B_i \leq 120$ . All  $A_i + B_i$  are the same.
- $1 \leq S_i \leq 10^6$ .

## Example

standard input	standard output
2	Case #1: 115.000000
1	Case #2: 135.000000
30 70	
15 15	
2	
30 15 70	
10 20	
20 10	

## Note

In the first test case, it takes Mr. Ang 30 seconds to the first traffic light, in the worst case he had to wait 15 seconds until it gets green. Then 70 seconds to the company. Total time is  $30 + 15 + 70 = 115$  seconds;

In the second test case, it still takes Mr. Ang 30 seconds to the first traffic light, as Mr. Ang could make  $OFF_1 = 0, OFF_2 = 25$ . If Mr. Ang meets red light at the first traffic light, he will definitely meet green light at the seconds one. So in the worst case, it takes Mr. Ang  $30 + 20 + 15 + 0 + 70 = 135$  seconds.

## Problem D. Mr. Panda and Geometric Sequence

Input file:       standard input  
Output file:      standard output

Mr. Panda likes playing with numbers. One day he picked up a number 3612 and found it's interesting. 3612 can be divided into 3 numbers 3, 6 and 12 which forms an integer geometric sequence.

An integer geometric sequence is a sequence of at least 3 positive integer numbers  $a_0, a_1, \dots, a_{n-1}$ , also there is a positive number  $D$  ( $D > 1$ ) that for each  $i$  ( $0 \leq i < n - 1$ ),  $a_i \times D = a_{i+1}$ .

Mr. Panda named this kind of numbers "Happy Number". He also announced that leading zeros are forbidden, which means there should be no extra zeros before the numbers. Now Mr. Panda would like to know how many Happy Numbers are between  $L$  and  $R$  inclusive.

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each test case contains one line which consists of two numbers  $L$  and  $R$ .

### 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 number of Happy Numbers between  $L$  and  $R$  inclusive.

### Limits

- $1 \leq T \leq 2 \times 10^5$ .
- $0 \leq L \leq R \leq 10^{15}$ .

### Example

standard input	standard output
4	Case #1: 1
123 124	Case #2: 1
468 470	Case #3: 1
248 248	Case #4: 1
124816 124832	

### Note

In the first test case, the only Happy Number between 123 and 124 is 124, in which  $1 \times 2 = 2$  and  $2 \times 2 = 4$ .

In the second test case, the only Happy Number is 248.

In the third test case, the only Happy Number is 469, the common ratio is  $\frac{3}{2}$ .

In the fourth test case, the only Happy number between 124816 and 124832 is 124816, in which  $1 \times 2 = 2$ ,  $2 \times 2 = 4$ ,  $4 \times 2 = 8$  and  $8 \times 2 = 16$ .

## Problem E. Snakebird

Input file:       standard input  
Output file:      standard output

Do you like puzzles, fruits and the notion of crossing a snake with a bird for no good reason? If the answer to at least one of those questions is yes, then Snakebird might be the game for you. Solve your way through this head-scratcher of a puzzle game and help the snakebirds sate their hunger.

The game starts with a board of  $M \times N$  cells, at the beginning each cell is one of the following types.

- ‘R’, ‘G’, ‘B’: Your snakebirds’ head. You’ll have at least 1 snakebird and at most 3 snakebirds on the board. Once you are moving a snakebird, other snakebirds are treat as obstacles. That means you can not enter a cell that is occupied by other snakebirds.
- ‘>’, ‘<’, ‘^’, ‘v’: Your snakebirds’ body is pointing to another body that is closer to the head. These 4 types are pointing right, left, up, down respective. It’s guaranteed that each head/body is pointed by exactly one other body except the last one. It’s also guaranteed that each body is pointing to a head/body.
- ‘.’: Empty cell. Your snakebirds can enter empty cells. When a snakebird enters an empty cell, it’s head goes into this cell and all it’s other bodies go into the position of previous head/body. This action finishes instantly.
- ‘#’: Obstacle. Your snakebirds can not enter obstacles, but if any piece of the snakebird is one cell above the obstacle, the obstacle supports the snakebird and the snakebird will stop falling.
- ‘|’: Spike. Your snakebirds can either enter spikes or falling into the spikes. However, if this happened, the snakebird dies.
- ‘@’: Fruit. Your snakebirds can enter the cell of fruit. When a snakebird enters a cell of fruit, the fruit disappears and the cell becomes an empty cell. Meanwhile, all head/body of the snakebird stay at the position and the snakebird has a new head in the cell of the fruit. The previous head becomes a body connecting with the new head. But when snakebird is falling the fruit is treated as an obstacle even the head of any snakebirds falling towards the fruit.
- ‘X’: Exit. There is exactly 1 exit on the board. Before all the fruits disappear, the exit is inactive and just like an empty cell. After **ALL** the fruits disappear, anytime a snakebird’s head is in the Exit cell, the snakebird disappears and goes into heaven. Even if the snakebird is falling. If a snakebird falls on a spike or falls out of the board while it’s head falls into the Exit cell, it dead before going to heaven. :(

On each move, you can select one of the snakebirds, and move it’s head to one of the adjacent cells(four directions) to enter an empty/fruit cell. After each move, something magic is happening, your snakebirds are subject to gravity, which means all snakebirds without support start falling until they land on some obstacles(or other snakebirds with support) or dead. Falling on spike or falling out of the board causes dead. The shape of the snakebird doesn’t change during the falling.

It’s guaranteed that nothing is going to fall at the beginning.

To help Redbird, Greenbird and Bluebird win the game, you need to send all snakebirds to heaven alive. Find out the minimal number of moves needed.

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

The first line of each test case contains 2 numbers  $M$ ,  $N$ , the rows and columns of the board. The following  $M$  lines, each line contains  $N$  characters representing the board.

## Output

For each test case, output one line containing “Case #x: y” in the one line. x is the test case number (starting from 1) and y is the minimum moves needed to send all snakebirds to heaven. If it’s impossible to do that, output -1 instead.

## Limits

- $1 \leq T \leq 10$ .
- $1 \leq N, M \leq 20$ .
- The total number of fruits, snakebirds head/body on a board is at most 10.
- It’s guaranteed that nothing is falling at the beginning.

## Example

standard input	standard output
<pre> 8 4 4 .... ...v X..G ##.# 4 4 .... ...v X..G #..# 4 4 ..X. .... .R.@ #### 4 7 ..... ..... &gt;&gt;B ..X ##### 4 7 ..... v..... &gt;&gt;B ..X ##### 4 5 &gt;R... ##... ##.X. ##... 4 5 &gt;R... ##... ## X. ##... 9 11 .....X... ..... ....#..... ..... .#..... .#..... .#.#...B&lt;&lt; ....#..G&lt;&lt;^ .....#### </pre>	<pre> Case #1: 3 Case #2: -1 Case #3: 6 Case #4: -1 Case #5: 6 Case #6: 2 Case #7: -1 Case #8: 46 </pre>

## Note

In the first test case, the green snakebird can go to left and enter the exit.

In the second test case, as the gap is too big, it's impossible to enter the exit.



In the third test case, the red snakebird needs to go right to eat the fruit, then enter the exit.

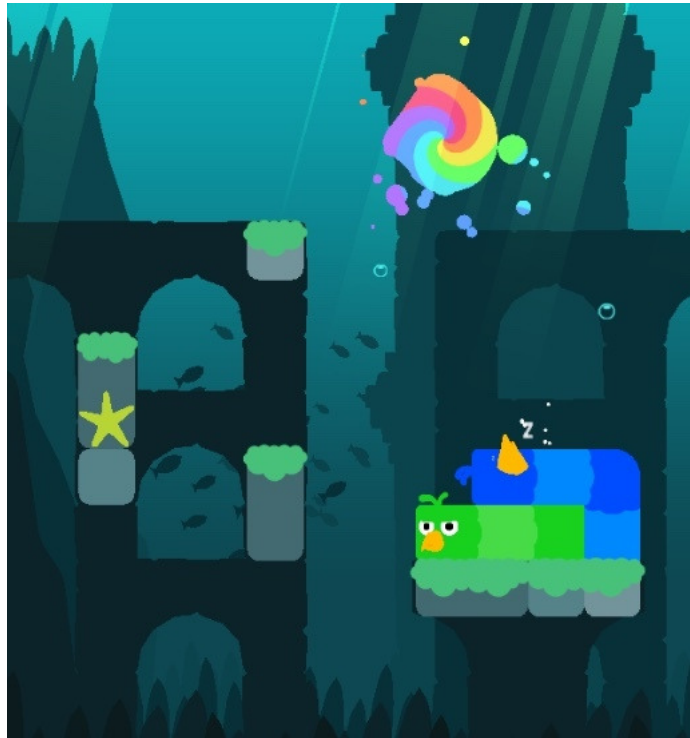
In the fourth test case, there is a spike in the middle, the snakebird of length 3 is not able to cross through.

In the fifth test case, the snakebird of length 4 is able to cross through.

In the sixth test case, the snakebird can just go right twice and jump to the exit.

In the seventh test case, the snakebird can also jump to the exit, but a spike kills it at the same time it reaches the exit.

The last sample looks like the picture below.



## Problem F. Good Number

Input file:        standard input  
Output file:       standard output

Given a positive integer  $M$  that consists of  $k$  digits in decimal notation without leading zeros, a ‘digit rotation’ of  $M$  means a number that is generated by swapping the first  $j$  digits and the rest  $(k - j)$  digits on  $M$ , for any  $j$  ( $0 \leq j < k$ ). For example, both 231 and 312 are digit rotations of 123, and neither 321 nor 132 is.

A positive integer  $G$  is a good number if any digit rotation of  $G$  does not exceed  $G$ .

Given a positive integer  $N$ , return the largest good number that does not exceed  $N$ .

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each line contains an integer  $N$  which is represented in decimal notation.

### 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 largest good number that does not exceed  $N$ .

### Limits

- $1 \leq T \leq 500$ .
- $1 \leq N \leq 10^{1000000}$ .
- $1 \leq \sum |N| \leq 3 \times 10^7$ .

### Example

standard input	standard output
4	Case #1: 110
110	Case #2: 10000
10010	Case #3: 44444
45363	Case #4: 8383837
8394634	

### Note

In the first sample case, 110 is a good number since all the digit rotations of it 101 and 011 are smaller than 110.

In the second sample case, 10010 itself is not a good number because one of the digit rotations 10100 is larger than 10010. 10001 to 10009 either, since 10001’s rotation is 11000, 10009’s rotation is 91000, similar for 10002 to 10008. Therefore, 10000 is the largest good number.

## Problem G. Image Recognition

Input file:        standard input  
Output file:       standard output

There are  $N$  binary images numbered from 0 to  $N - 1$ . Each image contains  $M$  pixels numbered from 0 to  $M - 1$ . Each pixel is either black or white. Images are pairwise different by at least one pixel.

Now there are some queries. Each query contains a subset of these images. For each query, you want to find a set of pixels that can distinguish every pair of images in it. Two images can be distinguished from each other by a given set of pixels if they have different pixel values in at least one of the given pixels. The number of pixels selected should be less than the number of images in each query.

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

The first line of each test case consists of three integers  $N$ ,  $M$ , and  $Q$ , indicating the number of images, the number of pixels in each image, and the number of queries. The following  $N$  lines describe the content of each image. Each line contains a 01 string of length  $M$  representing the content of the image. The following  $Q$  lines describe the queries. Each query starts with an integer  $K_i$  indicating the number of images of this query, then comes a list of space separated integers indicating the indices of the images.

### Output

For each test case, output one line containing "Case #x:", where  $x$  is the test case number (starting from 1), then follows  $Q$  lines. Each line starts with an integer indicating the number of pixels needed to distinguish the images in the query, then comes a list of space separated integers indicating the indices of the pixels. There could be multiple answers to each query. You may output any of them as long as the number of pixels selected is less than  $K_i$ .

### Limits

- $1 \leq T \leq 100$ .
- $1 \leq N \leq 2000$ .
- $1 \leq M \leq 4000$ .
- $1 \leq Q \leq 2000$ .
- $1 \leq K_i \leq N$ .
- The sum of  $N$  in all cases  $\leq 2 \times 10^4$ .
- The sum of  $K_i$  in all cases  $\leq 5 \times 10^6$ .

### Example

standard input	standard output
1	Case #1:
3 3 4	1 1
101	1 0
110	1 0
001	2 0 1
2 0 1	
2 0 2	
2 1 2	
3 0 1 2	

## Problem H. Mr. Panda and Birthday Song

Input file:        standard input  
Output file:       standard output

Mrs. Panda's birthday is coming. Mr. Panda wants to compose a song as gift for her birthday.

It is known that Mrs. Panda does not like a song if and only if its lyrics contain  $X$  vowels in a row, or  $Y$  consonants in a row (Otherwise, Mrs. Panda likes the song). The letters 'a', 'e', 'i', 'o', 'u' are vowels, and all others are consonants.

Though Mr. Panda is a gifted singer, he is bad at composing lyrics. Mr. Panda wants the song to be special. Thus, he searched Google for a template for lyrics of the song.

The template consists of lowercase English letters and possibly question marks. For example, "happybirthday" (without quotes, the same below), "????ybirthday" are valid templates but neither "happy birthday" nor "HappyBirthday" is. Mr. Panda needs to substitute all the question marks with lowercase English letters so that it becomes actual lyrics of the song.

Mr. Panda wants to give a surprise to Mrs. Panda. So, Mr. Panda hopes to compose not only a song from the template that Mrs. Panda likes but also a song from the same template that Mrs. Panda dislikes.

Because Mr. Panda is really bad at composing lyrics, even with a template, the task has confused him for days. Luckily, Mr. Panda knows you are in the contest and wants to ask you for help.

For a given template, output either "DISLIKE" (without quotes, the same below) if Mrs. Panda dislikes all the songs that are generated from the template (that means you cannot substitute letters for question marks so that Mrs. Panda likes the song), "LIKE" if Mrs. Panda likes all the songs that are generated from the template, or "SURPRISE" if Mr. Panda can compose a song Mrs. Panda likes and another song Mrs. Panda dislikes.

### Input

The first line of the input gives the number of test cases, an integer  $T$ .  $T$  test cases follow.

Each test case consists of a line containing a string  $S$ , the template that Mr. Panda gets, an integer  $X$ , the minimum number of continuous vowels that Mrs. Panda dislikes, and an integer  $Y$ , the minimum number of continuous consonants that Mrs. Panda dislikes. In the template  $S$ , each character can be either one of lowercase English letters ('a' to 'z') or question mark ('?').

### Output

For each test case, output one line containing "Case #x: y", where  $x$  is the test case number (starting from 1) and  $y$  can be either "LIKE", "DISLIKE" or "SURPRISE" as mentioned in the problem description.

### Limits

- $1 \leq T \leq 300$ .
- $2 \leq |S| \leq 10^6$ .
- $1 \leq X \leq |S|$ .
- $1 \leq Y \leq |S|$ .
- Sum of  $|S|$  over all test cases  $\leq 5 \times 10^7$ .

## Example

standard input	standard output
5	Case #1: DISLIKE
happybirthda? 3 4	Case #2: LIKE
happybirth?ay 3 5	Case #3: SURPRISE
happybirthd?y 3 5	Case #4: SURPRISE
hellow?rld 3 5	Case #5: DISLIKE
helllllowoorld 3 5	

## Problem I. PLAYERUNKNOWN'S BATTLEGROUNDS

Input file:       standard input  
Output file:      standard output

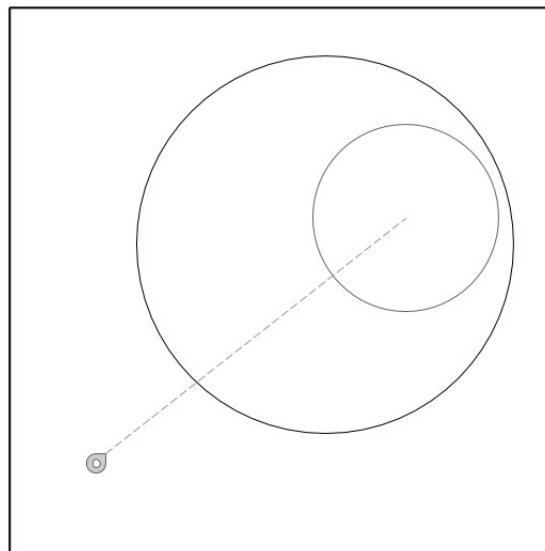
PLAYERUNKNOWN'S BATTLEGROUNDS (PUBG, also known as "Chicken Dinner") is the most popular survival shooter game in recent months. Players are dropped into a wide, open area, and they must fight with each other until the last person survives. Specifically, there is a shrinking circle called safe area in the battlefield, while players outside this area will suffer continued damage, which forces them to confront each other and creates chances of skirmish.

Some players are particularly stupid, so they became users of the WG company. But recently, WG company often receive complaints from users that they often die outside the safe area. After carefully study, WG company found that their users are too stupid to know how to run into the safe area. Therefore, they decided to assist their users to complete this action.

After creating a coordinate system on the map, the player is at  $(x_0, y_0)$  point at moment 0, with movement speed  $v$ . At this moment, there is a circle  $C_1$  with radius of  $r_1$  and center at  $(x_1, y_1)$  indicating the safe area now. And players outside the safe area will get  $d_1$  continued damage per unit of time. The safe area will start to shrink at time  $t_1$  and becomes a smaller circle  $C_2$  at time  $t_2$ , whose center is  $(x_2, y_2)$  and has radius  $r_2$ . Since then, players outside the safe area will get  $d_2$  damage per unit of time.

$C_2$  is strictly in  $C_1$ , and the change of circle radius and the movement of circle center are at a constant speed. However, damages outside the safe area will change from  $d_1$  to  $d_2$  instantly at  $t_2$ .

Your mission is to calculate the minimum damage the player incurs from the moment 0 till he/she reaches the final safe area  $C_2$  if the player moves optimally.



### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each test case contains three lines. The first line contains three integers  $x_0$ ,  $y_0$  and  $v$ , indicating the initial position and moving speed of player. The second line contains five integers  $x_1$ ,  $y_1$ ,  $r_1$ ,  $d_1$  and  $t_1$ , while the third line also contains five integers  $x_2$ ,  $y_2$ ,  $r_2$ ,  $d_2$  and  $t_2$ , as described above respectively.

## 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 damages.

y will be considered correct if it is within an absolute or relative error of  $10^{-4}$  of the correct answer.

## Limits

- $1 \leq T \leq 100$ .
- $0 \leq x_0 \leq 80000$ .
- $0 \leq y_0 \leq 80000$ .
- $50 \leq v \leq 400$ .
- Every point  $(x, y)$  on  $C_1$  has  $0 \leq x \leq 80000$  and  $0 \leq y \leq 80000$ .
- $C_2$  is strictly in  $C_1$ .
- $1 \leq d1 \leq d2 \leq 11$ .
- $0 \leq t1 \leq t2 \leq 80000$ .

## Example

standard input	standard output
3	Case #1: 4.2537599506
700 100 50	Case #2: 16.5388203202
600 600 600 5 0	Case #3: 28.4702942694
800 400 200 10 2	
700 0 80	
600 600 600 5 0	
800 400 200 10 2	
100 100 150	
600 600 600 5 1	
800 800 200 10 4	

## Problem J. Straight Master

Input file:        standard input  
Output file:       standard output

A straight is a poker hand containing five cards of sequential rank, not necessarily to be the same suit. For example, a hand containing 7 club, 6 spade, 5 spade, 4 heart and 3 diamond forms a straight. In this problem, we extend the definition of a straight to allow 3 to 5 cards of sequential rank. Hence a hand containing K spade, Q club, and J heart is also a straight.

Mr. Panda is playing a poker game called Straight Master. The game uses a large deck of card that has  $N$  ranks from 1 to  $N$ . The rule of the game is simple: split the cards in Mr. Panda's hand into several straights of length from 3 to 5.

Now given a hand of cards, can you help Mr. Panda to determine if it is possible to split the cards into straights?

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each test case contains two lines. The first line contains an integer  $N$ , indicating the number of ranks in the deck. The next line contains  $N$  integers  $a_1, a_2, \dots, a_N$  indicating the number of cards for each rank in Mr. Panda's hand.

### 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 Yes if Mr. Panda can split all his cards into straights of length from 3 to 5, or No otherwise.

### Limits

- $1 \leq T \leq 100$ .
- $1 \leq N \leq 2 \times 10^5$ .
- $0 \leq a_i \leq 10^9$ .
- $\sum N \leq 4 \times 10^6$ .

### Example

standard input	standard output
2	Case #1: Yes
13	Case #2: No
1 2 2 1 0 0 0 0 0 0 0 0 0	
13	
1 1 1 1 0 1 1 0 0 0 0 0 0	

### Note

In the first test case, Mr. Panda can split his cards into two straights:  $[1, 2, 3]$  and  $[2, 3, 4]$ . In the second test case, there is no way to form a straight for card 6 and 7.



## Problem K. Downgrade

Input file:        standard input  
Output file:       standard output

The God of Sheep plays an RPG game recently. The game has a level system that records a player's level with a pair of positive integers  $A$  and  $B$  where  $A$  is the major level and  $B$  is the minor level. For each major level  $k$ , you have to reach minor level  $L_k$  in order to upgrade to the next level  $k + 1$ . Different major levels may have different number of minor levels. For example, the God of Sheep is currently on level 3-4, and level 3 contains 4 minor levels: 3-1, 3-2, 3-3, and 3-4. Once the God of Sheep upgrades, he will be on level 4-1.

The God of Sheep has work to do now. Countless sheep are suffering in slaughter houses, woolsheds, and dairy farms. The God of Sheep has to rescue his fellow sheep citizens. He will be busy with the rescue plan for a couple of days and meanwhile the level of the game will be downgraded each day as a penalty. The downgrade penalty works like this: with each day the God of Sheep not playing the game, his level  $A$ - $B$  will firstly be transformed to  $A$  minor levels (with minor level  $B$  discarded), and then be cast to an equivalent major-minor pair as his new level. For example, suppose both level 1 and level 2 have 2 minor levels, and the God of Sheep is currently on level 3-3. For the first day of his absence, the level will become 2-1, the second day the level will become 1-2, the third day the level will become 1-1, and from the third day on the level will keep the same level as 1-1.

The God of Sheep wonders which level will he be when he comes back from the rescue?

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each test case contains two lines. The first line contains three integers:  $A$ ,  $B$  and  $N$ , indicating the current major level, minor level, and the number of days the God of Sheep will be away from the game. The next line contains  $A$  integers:  $L_1, L_2, \dots, L_A$  indicating the number of minor levels in each major level.

### Output

For each test case, output one line containing "Case #x: y-z", where x is the test case number (starting from 1), y and z are the major level and the minor level when the God of Sheep is back to the game.

### Limits

- $1 \leq T \leq 20$ .
- $1 \leq A \leq 10^5$ .
- $1 \leq B \leq L_A$ .
- $1 \leq L_i, N \leq 10^9$ .

### Example

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

## Problem L. SOS

Input file:        **standard input**  
Output file:       **standard output**

Mr. Panda and Mr. Sheep are playing a game on a  $1 \times N$  game board. Initially, all the cells are empty. Mr. Panda and Mr. Sheep take alternate moves and Mr. Panda moves first.

On each move, a player must fill an 'S' or 'O' into an empty cell. After the move, if there are 3 consecutive cells from left to right form the word "SOS", the player wins the game and the game finished.

If the board gets filled up without SOS appearing consecutively, no one wins the game, the game end with draw.

Now, if both Mr. Panda and Mr. Sheep play optimally, find out the result of the game.

### Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow.

Each test case contains one line consists of one number  $N$ , indicating the size of the game board.

### 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 game result if both players play optimally. "Panda" for Mr. Panda winning the game, "Sheep" for Mr. Sheep winning the game, "Draw" for draw game.

### Limits

- $1 \leq T \leq 1000$ .
- $1 \leq N \leq 1000$ .

### Example

standard input	standard output
2	Case #1: Draw
3	Case #2: Panda
7	

### Note

In the first test case, as there are only 3 cells, both players cannot win as Mr. Sheep only has one move and after his move, there is still one cell left empty. So Mr. Sheep is impossible to win, but it's easy for him to avoid Mr. Panda to win. He can either fill 'O' in the first or third cell, or fill 'S' in the second cell to make it a draw game.

In the second test case, Mr. Panda could place an 'S' on the 4-th cell. Then whatever Mr. Sheep moves, Mr. Panda could win.

## Problem M. World Cup

Input file:       standard input  
Output file:      standard output

The 2018 World Cup will be hosted in Russia. 32 national teams will be divided into 8 groups. Each group consists of 4 teams. In group matches, each pair (unordered) of teams in the group will have a match. Top 2 teams with the highest score in each group will advance to eighth-finals. Winners of each eighth-final will advance to quarter-finals. Then, the winners of each quarter-final will advance to semi-finals. Eventually, the World Champion will be the winner of the World Final which is played between the two winners of the semi-finals.

Each match is labeled with a match ID sequenced from 1 to 63, with group matches followed by eighth-final matches followed by quarter-final matches followed by semi-finals matches and finally the final match.

*Zhuojie* is going to watch the 2018 World Cup. Since the World Champion of ACM-ICPC is very rich, he decides to spend 0.01% of his daily salary to buy tickets. However, there are only match IDs on the tickets and the prices are missing. Can you calculate how much *Google* pays *Zhuojie* every workday? Note that *Zhuojie* can buy multiple tickets for one match.

### Input

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

Each test case contains 3 lines. The first line contains 5 integers, indicating the ticket price for group match, eighth-final match, quarter-final match, semi-final match and the final match. The second line contains one integer  $N$ , the number of tickets *Zhuojie* buys. The third line contains  $N$  integers, each indicating the match ID on the ticket.

### 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 daily salary of *Zhuojie*.

### Limits

- $1 \leq T \leq 100$ .
- $1 \leq N \leq 10^5$ .
- $1 \leq \text{match price} \leq 1000$ .
- $1 \leq \text{match ID} \leq 63$ .

### Example

standard input	standard output
1 11 12 13 14 15 2 1 49	Case #1: 230000