China Collegiate Programming Comtest

南阳理工学院 NAN YANG INSTITUTE OF TECHNOLOGY

2015年中国大学生程序设计竞赛

THE 2015 CHINA COLLEGIATE PROGRAMMING CONTEST

正式赛题目

主办单位：中国大学生程序设计竞赛协会

承办单位：南阳理工学院 NAN YANG INSTITUTE OF TECHNOLOGY

2015年10月18日

The 2015 China Collegiate Programming Contest

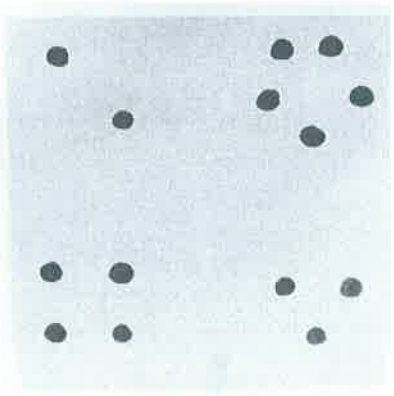
南陽理工學院
Nanyang Institute of Technology

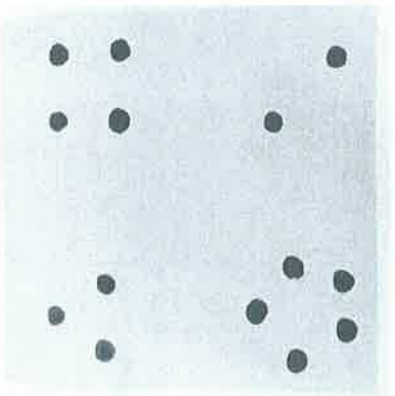
Problems

Problem A. Secrete Master Plan

Master Mind KongMing gave Fei Zhang a secrete master plan stashed in a pocket. The plan instructs how to deploy soldiers on the four corners of the city wall. Unfortunately, when Fei opened the pocket he found there are only four numbers written in dots on a piece of sheet. The numbers form a 2×2 matrix, but Fei didn't know the correct direction to hold the sheet. What a pity!

Given two secrete master plans. The first one is the master's original plan. The second one is the plan opened by Fei. As KongMing had many pockets to hand out, he might give Fei the wrong pocket. Determine if Fei receives the right pocket.





Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 10^4)$. T test cases follow. Each test case contains 4 lines. Each line contains two integers a_{i0} and a_{i1} $(1 \leq a_{i0}, a_{i1} \leq 100)$. The first two lines stands for the original plan, the 3_{rd} and 4_{th} line stands for the plan Fei opened.

Output

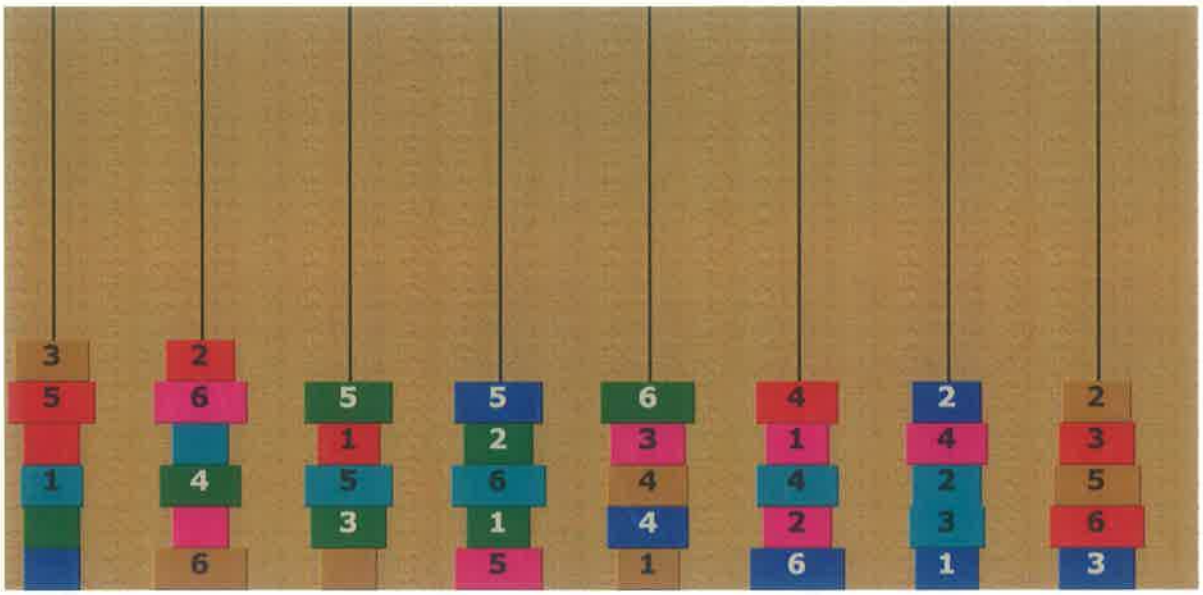
For each test case, output one line containing "`Case #x: y`", where x is the test case number (starting from 1) and y is either "`POSSIBLE`" or "`IMPOSSIBLE`" (quotes for clarity).

Sample input and output

Sample Input	Sample Output
4 1 2 3 4 1 2 3 4 1 2 3 4 3 1 4 2 1 2 3 4 3 2 4 1 1 2 3 4 4 3 2 1	Case #1: POSSIBLE Case #2: POSSIBLE Case #3: IMPOSSIBLE Case #4: POSSIBLE

Problem B. Build Towers

Build towers on a game board is very interesting. Here we introduce one kind of game which is also building towers on the game board.



There are n sticks on the game board. At the beginning of the game, we put plates of different colors on the sticks. There are $n-2$ different colors of plates. For each color, there are plates of size 0, 1, 2, 3, 4, 5, 6. One for each size. So totally there are $7(n-2)$ plates put on the sticks.

If a plate of size x is put right above the plate of size $x+1$ of the same color, they are glued together and not able to be split any more. In other words, they must be moved together in the rest of the game.

For each move, we can only move the top plate of each stick, maybe it's glued with several other plates, so they will be moved together. We can only move them onto a stick that the top plate is in the same color with the moving one and is larger than the moving plates. Or we can move them onto an empty stick.

The goal of this game is to make $n-2$ towers on any of the $n-2$ different sticks. Each of them are made of 7 plates in the order of 0, 1, 2, 3, 4, 5, 6 from top to bottom.

Now you need to find the minimal number of moved to achieve this.

Input

The first line of the input gives the number of test cases, $T(1 \le T \le 20)$. T test cases follow. Each test case starts with number $n(n = 8)$, the number of sticks on the game board. Each of the next n lines starts with a number $P_i(1 \le P_i \le 6)$ represents the number of plates set on each stick. P_i strings follow, each represents a plate set on that stick from bottom to top. The first character of the string represents the color and the rest represents the size or size range.

It's guaranteed that the number of colors equals $n-2$ and there are 7 different plates for each color, 0 to 6.

It's also guaranteed that there exists at least one solution for each test case.

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 number of moves to make $n-2$ complete towers.

Sample input and output

Sample Input	Sample Output
1 8 6 A0 B0 E1 C0 C5 F3 6 F6 D0 B4 E0 D6 C2 5 F0 B3 E5 C1 B5 5 D5 B1 E6 B2 A5 5 F1 A4 F4 D3 B6 5 A6 D2 E4 D1 C4 4 A1 E2-3 D4 A2 5 A3 C6 F5 C3 F2	Case #1: 47

Problem C. The Battle of Chibi

Cao Cao made up a big army and was going to invade the whole South China. Yu Zhou was worried about it. He thought the only way to beat Cao Cao is to have a spy in Cao Cao's army. But all generals and soldiers of Cao Cao were loyal, it's impossible to convince any of them to betray Cao Cao.

So there is only one way left for Yu Zhou, send someone to fake surrender Cao Cao. Gai Huang was selected for this important mission. However, Cao Cao was not easy to believe others, so Gai Huang must leak some important information to Cao Cao before surrendering.

Yu Zhou discussed with Gai Huang and worked out N information to be leaked, in happening order. Each of the information was estimated to has a_i value in Cao Cao's opinion.

Actually, if you leak information with strict increasing value could accelerate making Cao Cao believe you. So Gai Huang decided to leak exact M information with strict increasing value in happening order. In other words, Gai Huang will not change the order of the N information and just select M of them. Find out how many ways Gai Huang could do this.

Input

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

Each test case begins with two numbers $N(1 \leq N \leq 10^3)$ and $M(1 \leq M \leq N)$, indicating the number of information and number of information Gai Huang will select. Then N numbers in a line, the i_{th} number $a_i(1 \leq a_i \leq 10^9)$ indicates the value in Cao Cao's opinion of the i_{th} information in happening order.

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 ways Gai Huang can select the information.

The result is too large, and you need to output the result mod by $1000000007(10^9 + 7)$.

Sample input and output

Sample Input	Sample Output
2 3 2 1 2 3 3 2 3 2 1	Case #1: 3 Case #2: 0

Note

In the first cases, Gai Huang need to leak 2 information out of 3. He could leak any 2 information as all the information value are in increasing order. In the second cases, Gai Huang has no choice as selecting any 2 information is not in increasing order.

Problem D. Pick The Sticks

The story happened long long ago. One day, Cao Cao made a special order called "Chicken Rib" to his army. No one got his point and all became very panic. However, Cao Cao himself felt very proud of his interesting idea and enjoyed it.

Xiu Yang, one of the cleverest counselors of Cao Cao, understood the command Rather than keep it to himself, he told the point to the whole army. Cao Cao got very angry at his cleverness and would like to punish Xiu Yang. But how can you punish someone because he's clever? By looking at the chicken rib, he finally got a new idea to punish Xiu Yang.

He told Xiu Yang that as his reward of encrypting the special order, he could take as many gold sticks as possible from his desk. But he could only use one stick as the container.

Formally, we can treat the container stick as an L length segment. And the gold sticks as segments too. There were many gold sticks with different length a_i and value v_i. Xiu Yang needed to put these gold segments onto the container segment. No gold segment was allowed to be overlapped. Luckily, Xiu Yang came up with a good idea. On the two sides of the container, he could make part of the gold sticks outside the container as long as the center of the gravity of each gold stick was still within the container. This could help him get more valuable gold sticks.

As a result, Xiu Yang took too many gold sticks which made Cao Cao much more angry. Cao Cao killed Xiu Yang before he made himself home. So no one knows how many gold sticks Xiu Yang made it in the container.

Can you help solve the mystery by finding out what's the maximum value of the gold sticks Xiu Yang could have taken?

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 100)$. T test cases follow. Each test case start with two integers, $N(1 \leq N \leq 1000)$ and $L(1 \leq L \leq 2000)$, represents the number of gold sticks and the length of the container stick. N lines follow. Each line consist of two integers, $a_i(1 \leq a_i \leq 2000)$ and $v_i(1 \leq v_i \leq 10^9)$, represents the length and the value of the i_{th} gold stick.

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 value of the gold sticks Xiu Yang could have taken.

Sample input and output

Sample Input	Sample Output
4 3 7 4 1 2 1 8 1 3 7 4 2 2 1 8 4 3 5 4 1 2 2 8 9 1 1 10 3	Case #1: 2 Case #2: 6 Case #3: 11 Case #4: 3

Note

In the third case, assume the container is lay on x-axis from 0 to 5. Xiu Yang could put the second gold stick center at 0 and put the third gold stick center at 5, so none of them will drop and he can get total $2 + 9 = 11$ value. In the fourth case, Xiu Yang could just put the only gold stick center on any position of $[0, 1]$, and he can get the value of 3.

Problem E. Ba Gua Zhen

During the Three-Kingdom period, there was a general named Xun Lu who belonged to Kingdom Wu. Once his troop were chasing Bei Liu, he was stuck in the Ba Gua Zhen from Liang Zhuge. The puzzle could be considered as an undirected graph with N vertexes and M edges. Each edge in the puzzle connected two vertexes which were u_i and v_i with a length of w_i. Liang Zhuge had great interests in the beauty of his puzzle, so there were no self-loops and between each pair of vertexes, there would be at most one edge in the puzzle. And it was also guaranteed that there was at least one path to go between each pair of vertexes.

Fortunately, there was an old man named Chengyan Huang who was willing to help Xun Lu to hack the puzzle. Chengyan told Xun Lu that he had to choose a vertex as the start point, then walk through some of the edges and return to the start point at last. During his walk, he could go through some edges any times. Since Liang Zhuge had some mysterious magic, Xun Lu could hack the puzzle if and only if he could find such a path with the maximum `XOR` sum of all the edges length he has passed. If the he passed some edge multiple times, the length would also be calculated by multiple times. Now, could you tell Xun Lu which is the maximum `XOR` circuit path in this puzzle to help him hack the puzzle?

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 30)$. T test cases follow.

Each test case begins with two integers $N(2 \leq N \leq 5 \times 10^4)$ and $M(1 \leq M \leq 10^5)$ in one line. Then M lines follow. Each line contains three integers u_i, v_i and $w_i(1 \leq u_i, v_i \leq N$, $0 \leq w_i \leq 2^{60} - 1)$ to describe all the edges in the puzzle.

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 `XOR` sum of one circuit path in the puzzle.

Sample input and output

Sample Input	Sample Output
2 3 3 1 2 1 1 3 2 2 3 0 6 7 1 2 1 1 3 1 2 3 1 3 4 4 4 5 2 4 6 2 5 6 2	Case #1: 3 Case #2: 3

Note

A `XOR` takes two bit patterns of equal length and performs the logical exclusive `OR` operation on

each pair of corresponding bits. The result in each position is 1 if only the first bit is 1 or only the second bit is 1, but will be 0 if both are 0 or both are 1. In this we perform the comparison of two bits, being 1 if the two bits are different, and 0 if they are the same.

Problem F. The Battle of Guandu

In the year of 200, two generals whose names are Cao Cao and Shao Yuan are fighting in Guandu. The battle of Guandu was a great battle and the two armies were fighting at M different battlefields whose numbers were 1 to M. There were also N villages nearby numbered from 1 to N. Cao Cao could train some warriors from those villages to strengthen his military. For village i, Cao Cao could only call for some number of warriors join the battlefield x_i. However, Shao Yuan's power was extremely strong at that time. So in order to protect themselves, village i would also send equal number of warriors to battlefield y_i and join the Yuan Shao's Army. If Cao Cao had called for one warrior from village i, he would have to pay c_i units of money for the village. There was no need for Cao Cao to pay for the warriors who would join Shao Yuan's army. At the beginning, there were no warriors of both sides in every battlefield.

As one of greatest strategist at that time, Cao Cao was considering how to beat Shao Yuan. As we can image, the battlefields would have different level of importance w_i. Some of the battlefields with $w_i = 2$ were very important, so Cao Cao had to guarantee that in these battlefields, the number of his warriors was greater than Shao Yuan's. And some of the battlefields with $w_i = 1$ were not as important as before, so Cao Cao had to make sure that the number of his warriors was greater or equal to Shao Yuan's. The other battlefields with $w_i = 0$ had no importance, so there were no restriction about the number of warriors in those battlefields. Now, given such conditions, could you help Cao Cao find the least number of money he had to pay to win the battlefield?

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 30)$. T test cases follow.

Each test case begins with two integers N and $M(1 \leq N, M \leq 10^5)$ in one line.

The second line contains N integers separated by blanks. The i_{th} integer $x_i(1 \leq x_i \leq M)$ means Cao Cao could call for warriors from village i to battlefield x_i.

The third line also contains N integers separated by blanks. The i_{th} integer $y_i(1 \leq y_i \leq M)$ means if Cao Cao called some number of warriors from village i, there would be the same number of warriors join Shao Yuan's army and fight in battlefield y_i.

The next line contains N integers separated by blanks. The i_{th} integer $c_i(0 \leq c_i \leq 10^5)$ means the number of money Cao Cao had to pay for each warrior from this village.

The last line contains M integers separated by blanks. The i_{th} number $w_i(w_i \in \{0, 1, 2\})$ means the importance level of ith battlefield.

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 least amount of money that Cao Cao had to pay for all the warriors to win the battle. If he couldn't win, $y = -1$.

Sample input and output

Sample Input	Sample Output
2 2 3 2 3 1 1 1 1 0 1 2 1 1 1 1 1 2	Case #1: 1 Case #2: -1

Problem G. Ancient Go

Yu Zhou likes to play "Go" with Su Lu. From the historical research, we found that there are much difference on the rules between ancient go and modern go.

Here is the rules for ancient go they were playing:

- The game is played on a 8×8 cell board, the chess can be put on the intersection of the board lines, so there are 9×9 different positions to put the chess.
- Yu Zhou always takes the black and Su Lu the white. They put the chess onto the game board alternately.
- The chess of the same color makes connected components(connected by the board lines), for each of the components, if it's not connected with any of the empty cells, this component dies and will be removed from the game board.
- When one of the player makes his move, check the opponent's components first. After removing the dead opponent's components, check with the player's components and remove the dead components.

One day, Yu Zhou was playing ancient go with Su Lu at home. It's Yu Zhou's move now. But they had to go for an emergency military action. Little Qiao looked at the game board and would like to know whether Yu Zhou has a move to kill at least one of Su Lu's chess.

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 100)$. T test cases follow. Test cases are separated by an empty line. Each test case consist of 9 lines represent the game board. Each line consists of 9 characters. Each character represents a cell on the game board. '.' represents an empty cell. 'x' represents a cell with black chess which owned by Yu Zhou. 'o' represents a cell with white chess which owned by Su Lu.

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 "`Can kill in one move!!!`" if Yu Zhou has a move to kill at least one of Su Lu's components. "`Can not kill in one move!!!`" otherwise.

Sample input and output

Sample Input	Sample Output
2 .....xo ....x..... .xox....x .o.o...xo ..o..... .....xxxo ....xooo. .....ox. .....o. ...o..... ..o.o.... ...o..... ....o. ...x..... .....o	Case #1: Can kill in one move!!! Case #2: Can not kill in one move!!!

Note

In the first test case, Yu Zhou has 4 different ways to kill Su Lu's component.

In the second test case, there is no way to kill Su Lu's component.

Problem H. Sudoku

Yi Sima was one of the best counselors of Cao Cao. He likes to play a funny game himself. It looks like the modern Sudoku, but smaller.

Actually, Yi Sima was playing it different. First of all, he tried to generate a 4×4 board with every row contains 1 to 4, every column contains 1 to 4. Also he made sure that if we cut the board into four 2×2 pieces, every piece contains 1 to 4.

Then, he removed several numbers from the board and gave it to another guy to recover it. As other counselors are not as smart as Yi Sima, Yi Sima always made sure that the board only has one way to recover.

Actually, you are seeing this because you've passed through to the Three-Kingdom Age. You can recover the board to make Yi Sima happy and be promoted. Go and do it!!!

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 100)$. T test cases follow. Each test case starts with an empty line followed by 4 lines. Each line consist of 4 characters. Each character represents the number in the corresponding cell (one of '1', '2', '3', '4'). '*' represents that number was removed by Yi Sima.

It's guaranteed that there will be exactly one way to recover the board.

Output

For each test case, output one line containing "`Case #x:`", where x is the test case number (starting from 1). Then output 4 lines with 4 characters each. indicate the recovered board.

Sample input and output

Sample Input	Sample Output
3 **** 2341 4123 3214 *243 *312 *421 *134 *41* **3* 2*41 4*2*	Case #1: 1432 2341 4123 3214 Case #2: 1243 4312 3421 2134 Case #3: 3412 1234 2341 4123

Problem I. Mahjong

The game of Mahjong originated in China and has become popular around the world. You do not need to have prior experience with Mahjong to solve this problem, and we will use different rules.

In our version of Mahjong, the player is given a set of $4K$ tiles. Each tile has an integer rank written on it, and there are four identical copies of each rank from 1 to K. For example, for $K = 5$, the set of tiles would be: 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5.

The player's goal is to select M tiles from this set to form a winning hand. A *winning* hand consists of some number (possibly zero) of *triples* plus **exactly** one pair. A pair must consist of two tiles of the same rank. A triple can be either three tiles of the same rank (e.g., "2 2 2"), or three tiles with consecutive ranks (e.g., "3 4 5"). The ranks do not wrap around – for example, "4 5 1" is not a valid triple.

Given K and M, how many different winning hands are there? Two winning hands are considered the same if they use the same set of tiles, regardless of how those tiles are grouped to make triples and the pair. For instance, for $K = 4$, $M = 8$, the following two hands are considered the same:

"1 2 3, 1 2 3, 4 4"

"1 1, 2 3 4, 2 3 4"

Input

The first line of the input gives the number of test cases, $T(1 \le T \le 100)$. T lines follow. Each line contains two space-separated integers, $K(1 \le K \le 200)$ and $M(2 \le M \le min(200, 4K)$ and $M \equiv 2 \pmod 3$).

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 winning hands, modulo 1000000007 ($10^9 + 7$).

Sample input and output

Sample Input	Sample Output
4 1 2 3 5 4 5 9 14	Case #1: 1 Case #2: 9 Case #3: 20 Case #4: 13259

Note

In Case #1, there are only four tiles – 1, 1, 1, 1 – and the winning hand must consist of just a pair (with no triples). There is only one possibility – "1 1". (Note that all the '1's are interchangeable and it doesn't matter which two you pick.)

In Case #2, there are twelve tiles – 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3 – and the winning hand must consist of one triple and one pair. The nine possible hands are:

1 1 1, 2 2
1 1 1, 3 3

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

Note that "`3 3, 1 1 1`" would not be considered a different hand from "`1 1 1, 3 3`" – only the set of tiles matters, not how they are arranged.

Problem J. Walk Around The Campsite

Gan Jiang was one of Yu Zhou's friends but worked for Cao Cao. One day Gan Jiang visited Yu Zhou's army and would like to steal something from Yu Zhou's army in the midnight. Yu Zhou invited him to stay for the whole night and set up guards.

Gan Jiang saw there are N forts numbered 0, 1, 2, $\cdots$, $n-1$ in Yu Zhou's army, the guards are connecting the neighbours forts, including $(0,1)$, $(1,2)\cdots(n-2,n-1)$, $(n-1,0)$ to form a simple polygon. Gan Jiang was in the fort 0 only allowed to walk towards the higher numbered forts in straight line. He was not allowed to go inside of the polygon even pass by other vertices, but he can go along the edges of the polygon, e.g, go from fort 2 to fort 3. When Gan Jiang arrived a fort, he could steal letters of value V_i. But walking makes him unhappy and for each unit length walking, he will get 1 unit of unhappiness, Gan Jiang could end this mission at any time and ride a horse back to Cao Cao.

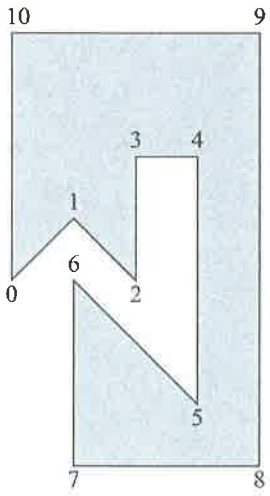


Figure 1: 3_{rd} sample

Gan Jiang would like to make the total value of letters as big as possible, but make the unhappiness as low as possible. So his target is to make the total value of letters minus total unhappiness as big as possible. Help Gan Jiang to get it.

Input

The first line of the input gives the number of test cases, $T(1 \leq 10)$. T test cases follow. Each test case starts with an integer $N(1 \leq N \leq 1000$. Each of the next N lines consists of 3 float numbers with 4 digits after decimal point, x_i, $y_i(-10^5 \leq x_i, y_i \leq 10^5)$, $V_i(0 \leq V_i \leq 10^5)$, (x_i, y_i) represents the coordinate of fort i and V_i represents the value of the letters in for i.

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 maximal value of total value of letters minus total unhappiness. y will be considered correct if it is within an absolute or relative error of 10^{-4} of the correct answer.

Sample input and output

Sample Input	Sample Output
3 4 0.0000 0.0000 0.0000 1.0000 0.0000 0.5000 1.0000 1.0000 2.0000 0.0000 1.0000 0.5000 5 0.0000 0.0000 0.0000 1.0000 1.0000 0.5000 2.0000 0.0000 2.0000 2.0000 2.0000 3.0000 0.0000 2.0000 0.0000 11 0.0000 0.0000 100.0000 1.0000 1.0000 100.0000 2.0000 0.0000 100.0000 2.0000 2.0000 100.0000 3.0000 2.0000 100.0000 3.0000 -2.0000 100.0000 1.0000 0.0000 100.0000 1.0000 -3.0000 100.0000 4.0000 -3.0000 100.0000 4.0000 4.0000 100.0000 0.0000 4.0000 100.0000	Case #1: 0.5000 Case #2: 1.0000 Case #3: 1070.3431

Problem K. Game Rooms

Your company has just constructed a new skyscraper, but you just noticed a terrible problem: there is only space to put one game room on each floor! The game rooms have not been furnished yet, so you can still decide which ones should be for table tennis and which ones should be for pool. There must be at least one game room of each type in the building.

Luckily, you know who will work where in this building (everyone has picked out offices). You know that there will be T_i table tennis players and P_i pool players on each floor. Our goal is to minimize the sum of distances for each employee to their nearest game room. The distance is the difference in floor numbers: 0 if an employee is on the same floor as a game room of their desired type, 1 if the nearest game room of the desired type is exactly one floor above or below the employee, and so on.

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 100)$. T test cases follow. Each test case begins with one line with an integer $N(2 \leq N \leq 4000)$, the number of floors in the building. N lines follow, each consists of 2 integers, T_i and $P_i(1 \leq T_i, P_i \leq 10^9)$, the number of table tennis and pool players on the i_{th} floor. The lines are given in increasing order of floor number, starting with floor 1 and going upward.

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 sum of distances.

Sample input and output

Sample Input	Sample Output
1 2 10 5 4 3	Case #1: 9

Note

In the first case, you can build a table tennis game room on the first floor and a pool game room on the second floor. In this case, the 5 pool players on the first floor will need to go one floor up, and the 4 table tennis players on the second floor will need to go one floor down. So the total distance is 9.

Problem L. Huatuo's Medicine

Huatuo was a famous doctor. He use identical bottles to carry the medicine. There are different types of medicine. Huatuo put medicines into the bottles and chain these bottles together.

However, there was a critical problem. When Huatuo arrived the patient's home, he took the chain out of his bag, and he could not recognize which bottle contains which type of medicine, but he remembers the order of the bottles on the chain.

Huatuo has his own solution to resolve this problem. When he need to bring 2 types of medicines, E.g. `A` and `B`, he will put `A` into one bottle and put `B` into two bottles. Then he will chain the bottles in the order of '`-B-A-B-`'. In this way, when he arrived the patient's home, he knew that the bottle in the middle is medicine `A` and the bottle on two sides are medicine `B`.

Now you need to help Huatuo to work out what's the minimal number of bottles needed if he want to bring N types of medicine.

Input

The first line of the input gives the number of test cases, $T(1 \leq T \leq 100)$. T lines follow. Each line consist of one integer $N(1 \leq N \leq 100)$, the number of types of the medicine.

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 number of bottles Huatuo needed.

Sample input and output

Sample Input	Sample Output
1 2	Case #1: 3