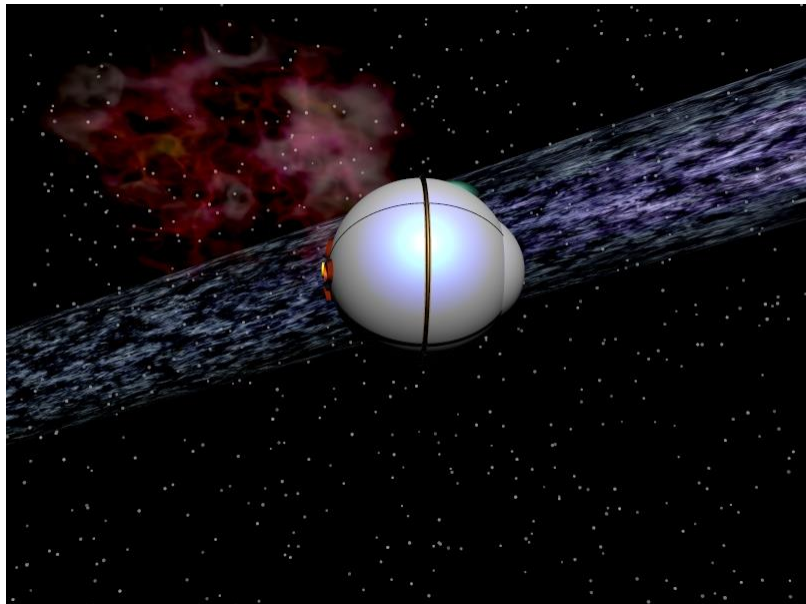


Problem G: SpaceRecon Tournament

SpaceRecon, the hottest game of 2011, is a real-time strategy thriller where players control your armies to destroy their opponent. Players may choose from one of the three available races -- Xurks (biological), Protoast (technological), and Earthians (humanoids) -- to build their respective armies by collecting resources from the land and spending them on army units, upgrades, and infrastructure. The player who destroys or outlasts their opponent is victorious.

The game has an intricate single-player story mode where players recreate scenes of the survival of Earthian Commander John Rainard's travels through the Xurkling planet, and also the charades of once-Earthian-now-Xurkling Queen Stephanie Karpenter. After completing the 15 hours of single player gameplay, most users try their hand at multiplayer head-to-head battles online using Actionweb, the number one SpaceRecon game matching hub.



Actionweb hosts tournaments of 2^M players and publishes the results of each tournament by listing each player handle and the number of match victories in the tournament. Tournaments are series of head-to-head matches between two players, the winner of the round advancing to the next round. The first R rounds are best of three (i.e., win two matches to win the round, any unnecessary games are not played), the remaining rounds are best of five (i.e., win three matches to win the round, any unnecessary games are not played). Each tournament has a different value of R , but is not published.

Input Format

The first line is an integer N ($1 \leq N \leq 100$), the number of test cases, which follow. Each test case begins with a line containing an integer M ($1 \leq M \leq 10$). The following 2^M lines are of format "player_handle number_of_match_victories". Player handles are alphanumeric and between 1 and 16 characters long.

You may assume that the data provided describes a valid tournament.

Output Format

Print the player handles sorted in descending order of which round they survived to. For players who survived the same number of rounds, sort them lexicographically by player handle.

Sample Input

```
1
2
John 1
Jake 5
Joe 4
Jane 0
```

Sample Output

```
Jake
Joe
Jane
John
```

Sean McIntyre
ACPC 2011



2223 - Disk Tree

Europe - Northeastern - 2000/2001

Hacker Bill has accidentally lost all the information from his workstation's hard drive and he has no backup copies of its contents. He does not regret for the loss of the files themselves, but for the very nice and convenient directory structure that he had created and cherished during years of work. Fortunately, Bill has several copies of directory listings from his hard drive. Using those listings he was able to recover full paths (like "WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86") for some directories. He put all of them in a file by writing each path he has found on a separate line. Your task is to write a program that will help Bill to restore his state of the art directory structure by providing nicely formatted directory tree.

Input

Input consists of several datasets. The first line of each dataset contains single integer number N ($1 \leq N \leq 500$) that denotes a total number of distinct directory paths. Then N lines with directory paths follow. Each directory path occupies a single line and does not contain any spaces, including leading or trailing ones. No path exceeds 80 characters. Each path is listed once and consists of a number of directory names separated by a back slash ("\").

Each directory name consists of 1 to 8 uppercase letters, numbers, or the special characters from the following list: exclamation mark, number sign, dollar sign, percent sign, ampersand, apostrophe, opening and closing parenthesis, hyphen sign, commercial at, circumflex accent, underscore, grave accent, opening and closing curly bracket, and tilde ("!#\$%&'()-@^_`{ }~").

There is a blank line after each dataset.

Output

For each test case, write to the output the formatted directory tree. Each directory name shall be listed on its own line preceded by a number of spaces that indicate its depth in the directory hierarchy. The subdirectories shall be listed in lexicographic order immediately after their parent directories preceded by one more space than their parent directory. Top level directories shall have no spaces printed before their names and shall be listed in lexicographic order. See sample below for clarification of the output format.

Print a blank line after each dataset

Sample Input

```
7
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86
```

Sample Output

```
GAMES
  DRIVERS
HOME
WIN
  SOFT
WINNT
  DRIVERS
  SYSTEM32
    CERTSRV
      CERTCO~1
        X86
      CONFIG
```

Northeastern 2000-2001

985 Round and Round Maze

You have been blindfolded and brought to a strange complex of mazes. Each maze is divided in squares, each one with a strange circular plate with arrows on it. The start of each maze is on the upper left corner and the exit is always on the bottom right. The following figure shows an example maze:

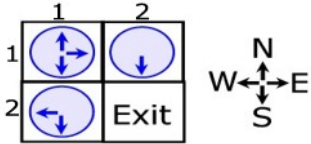


Figure 1: An example maze

During a single time unit you can go from a square to one of the four adjacent squares. But you can only follow the directions that the arrows on your current square point. Trying to do other thing will get you killed. Each time you pass from one square to another, all plates rotate 90 degrees in clockwise direction, changing the way the maze looks. Is it possible to get to the exit? And what is the best and fastest way to do that?

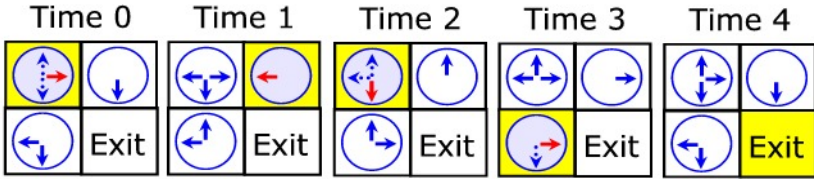


Figure 2: The quickest path to get out of the example maze

Given a particular maze in the conditions described above, your task is to discover how much time does it take the quickest path from the upper left corner to the bottom right corner (the exit). You must also discover if that is not possible.

Input

The input file contains several test cases, each of them as describes below.

The input will start with a single line containing two numbers separated by a single space: R and C indicating respectively the number of rows and columns of the maze ($2 \leq R, C \leq 500$).

Then there are exactly $(R * C) - 1$ lines indicating in which directions are the arrows of each square pointing. These lines are given in a specific order, starting from the north to the south, and from the west to the east. This is, if we use the notation (row,column), the lines are given in the order $(1, 1), (1, 2), \dots, (1, C), (2, 1), \dots, (2, N), \dots, (R, 1), \dots, (R, C - 1)$. The bottom right corner (R, C) is not given, since it is always the exit.

Each of these lines contains a single string of length one to four chars, indicating the arrows of the plate on time 0. The chars belong to the set N,S,W,E and represent respectively an arrow pointing to North, South, West and East. See example input 1 for a representation of the example maze given in figure 1. There will not be repeated chars on the same line and the chars can appear in any order.

Output

For each test case, the output should contain a single line with an integer that represents the time taken by the quickest path from the start (always square (1,1)) to the exit (always (R,C)). Remember that the plates always rotate when you change your current square (therefore it does not help to stay on the same place waiting for a rotation - it won't happen!) and you can only follow the directions that the arrows point on the present time you are on that square.

If there is no path from the start to the exit you should print 'no path to exit'.

Note:

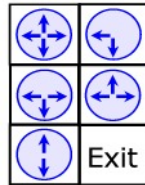


Figure 3: The maze of the sample input 2

Sample Input

```
2 2
NES
S
WS
3 2
NSWE
SW
SEW
NEW
SN
```

Sample Output

```
4
no path to exit
```

12707 Block Meh

Everybody knows Korimbai. Though he does not have time for any one. He is a legend in his own.

Now once Korimbai got very angry with the puny humans around him. He decided to block them on his social networking website Korimboi. (Korimboi is very popular website among puny humans, every one of you have 1 or more account on that, though people do not know about the actual creator (Korimbai) / actual name (Korimboi) of the site, because as always Korimbai has no time for that.)

Korimbai is a strange man, so he wants to block them in a strange way. As the creator of korimboi he has this special power of blocking more than 1 people at 1 click. He does that in the following way.

He chooses one puny human given that puny human's entrance and exit time be S_1 and E_1 respectively in Korimadda (Chat client in Korimboi) and selects it for blocking. Then he finds another puny human whose entrance time and exit time be S_2 and E_2 respectively. He can select it if and only if $S_2 > S_1$ and $E_2 < E_1$. Now he search for another puny human whose entrance time and exit time be S_3 and E_3 respectively and $S_3 > S_2$ and $E_3 < E_2$. He continues this search until there are nobody available in those criteria that $S_i > S_{i-1}$ and $E_i < E_{i-1}$. Then he blocks all of them at once in one click.

Now Korimbai wants to give this puzzle to you for your brain exercise (of course he knows the answer. He is THE Korimbai, but he has no time for that). He wants to know that given all the entrance and exit time for the puny humans in his korimadda list what is the minimum number of click Korimbai needs to give to delete all of the puny humans of his list.

Input

First line of input will contain the number of test cases, $T \leq 20$ to follow. In each test case 1-st line contains N , number of puny humans in Korimbai's lists. N line follows each containing 2 integers S_i and E_i , entrance and exit time for the i -th puny human

- $N \leq 20000$
- $0 \leq S_i \leq E_i \leq 1000000$

Output

For each input, print the output in the format, 'Case X : Y ' (here, X is the serial of the input and Y is the minimum number of click Korimbai needs to give to delete all of the puny humans of his list.

Output Explanation

On the first test case Korimbai can block 1st \rightarrow 4th puny human in 1 click and 2nd in 1 click and 3rd in 1 click totaling 3 clicks. He cannot do less than that.

On the second test case Korimbai can block 1st \rightarrow 2nd \rightarrow 4th puny human in 1 click and 3rd in 1 click totaling 2 clicks. He cannot do less than that.

Sample Input

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

3 4
4
1 5
2 4
3 5
3 3

Sample Output

Case 1: 3
Case 2: 2

Problem L

Goldbach and Euler

Input: standard input
Output: standard output
Time Limit: 40 seconds
Memory Limit: 40 MB

“That every number which is resolvable into two prime numbers can be resolved into as many prime numbers as you like, can be illustrated and confirmed by an observation which you have formerly communicated to me, namely that every even number is a sum of two primes, and since $(n-2)$ is also a sum of two prime numbers, n must be a sum of three, and also four prime numbers, and so on. If, however, n is an odd number, then it is certainly a sum of three prime numbers, since $(n-1)$ is a sum of two prime numbers, and can therefore be resolved into as many prime numbers as you like. However, that every number is a sum of two primes, I consider a theorem which is quite true, although I cannot demonstrate it.”

-- *Euler to Goldbach, 1742*

The above conjecture about all numbers being the sum of two primes (where **1** counts as a prime) is not always true, but it is more true for even numbers. Your task is to test the conjecture for specified integers, considering that prime numbers are the numbers which are positive and divisible by exactly two positive integers. Your program must be very efficient.

Input

The input file contains **100000** lines of input. Each line contains a single integer n ($0 < n \leq 100000000$). Input is terminated by end of file.

Output

For each line of input produce one line of output. This line should be of one of the following types:

```
n is not the sum of two primes!  
n is the sum of p1 and p2.
```

For the second case, always make sure that $(p2-p1)$ is positive and minimized.

Sample Input

```
11  
12
```

Sample Output

```
11 is not the sum of two primes!  
12 is the sum of 5 and 7.
```

(The Joint Effort Contest, Problem setter: Shahriar Manzoor, idea from a Waterloo Problem)

12726 One Friend at a Time

Bob is in love with Alice, a girl at his school. He always dreams of dating her, kissing her and playing weird games with her. But he is too shy to talk to Alice, so he decided to approach her through the internet first. He will start by adding her as a friend at LifeInvader (a very popular social network). But people often find it weird to accept you as a friend if they don't know you, except when you have at least K friends in common with them. So Bob has to add some of Alice's friends first. To add them, he has to add some of their friends too, and it goes on.

He finds it futile to have friends he doesn't really know at LifeInvader, so he wants to add the minimum additional friends. Help him know the minimum number of friends he has to add to be able to add Alice. It's possible that Bob is friends with Alice already in LifeInvader, and is just messing with you.

Input

The first line contains T ($T \leq 200$) — the number of test cases, after this line T test cases follows. Each test case starts with three integers N ($2 \leq N \leq 20$), M ($0 \leq M \leq \frac{N*(N-1)}{2}$) and K ($0 \leq K \leq N$) — the number of people in LifeInvader (including Bob and Alice), the number of current friendships and the minimum number of common friends you must have with someone to be able to add that person as a friend, correspondingly. Each person at LifeInvader is identified by a unique number between 1 and N . Bob is 1 and Alice is N . Then, M lines follow, each one with two numbers A and B ($1 \leq A, B \leq N$, $A \neq B$), meaning that persons A and B are currently friends.

Output

For each test case print a line containing 'Case #X: Y', where X is the case number, starting at 1, and Y is the minimum number of friends Bob has to add to be able to be friends with Alice, or '-1' if that is impossible.

Sample Input

```

3
3 2 1
1 2
2 3
7 6 1
1 2
2 3
3 4
4 5
5 6
6 7
4 3 2
1 2
2 4
3 4

```

Sample Output

Case #1: 0
Case #2: 4
Case #3: -1



4329 - Ping pong

Asia - Beijing - 2008/2009

N ($3 \leq N \leq 20000$) ping pong players live along a west-east street (consider the street as a line segment). Each player has a unique skill rank. To improve their skill rank, they often compete with each other. If two players want to compete, they must choose a referee among other ping pong players and hold the game in the referee's house. For some reason, the contestants can't choose a referee whose skill rank is higher or lower than both of theirs. The contestants have to walk to the referee's house, and because they are lazy, they want to make their total walking distance no more than the distance between their houses. Of course all players live in different houses and the position of their houses are all different. If the referee or any of the two contestants is different, we call two games different. Now is the problem: how many different games can be held in this ping pong street?

Input

The first line of the input contains an integer T ($1 \leq T \leq 20$), indicating the number of test cases, followed by T lines each of which describes a test case.

Every test case consists of $N + 1$ integers. The first integer is N , the number of players. Then N distinct integers a_1, a_2, \dots, a_N follow, indicating the skill rank of each player, in the order of west to east ($1 \leq a_i \leq 100000, i = 1 \dots N$).

Output

For each test case, output a single line contains an integer, the total number of different games.

Sample Input

```
1
3 1 2 3
```

Sample Output

```
1
```

Beijing 2008-2009

E

Component Placement

Input: Standard Input
Output: Standard Output

In circuit design, component placement is an important step in the design automation. Inferior placements may affect the performance and manufacturing cost.

You are given a PCB (Printed Circuit Board). It's a large green board. You can place components on either side of the PCB. However, cost of placing a component on both sides are not the same. You are given N components. For each component c_i , cost of placing it on the top layer is X_i and on the bottom layer is Y_i .

These components may interact with each other. If both the components are on the same side of the PCB, the interconnection cost is negligible. But, if they are on different sides, their interconnection is costly. For each such interconnection j , the cost will be Z_j .

Finally, some design issues restricts some components to be on the top side or bottom side. Now, find the minimum cost to place the components.

Input

First line contains a positive integer T ($T \leq 50$) that denotes the number of test cases.

Each test case starts with 2 integers N ($1 \leq N \leq 200$) and M ($0 \leq M \leq 100000$, $M \leq N * (N-1) / 2$), the number of components and number of interconnections. This will be followed by N integers in a line, each between 1 and 10000000 (inclusive), where i -th of it describes the cost of placing the component on the top layer. The next line contains N more integers, each between 1 and 10000000 (inclusive), where i -th of it denotes the cost of placing it on the bottom layer. The next line contains N more integers, each will be either 0, -1 or +1, where

- -1 means i -th component can only be placed on the bottom
- +1 means i -th component can only be placed on the top
- 0 means the component can be placed on either side

Then there will be M lines, each containing three integers, p , q , and r ($1 \leq p, q \leq N$, $1 \leq r \leq 10000000$), denoting that, p and q -th component has to be interconnected and if they are on different layers, the cost of interconnection will be r . There will be at most one interconnection between any pair or components.

Output

For each test case, output the minimum cost to place the components.

Sample Input

Output for Sample Input

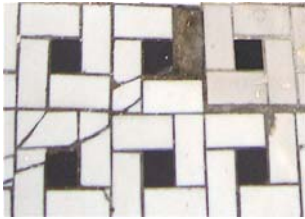
5	22
4 0	24
5 6 7 8	25
8 7 6 5	26
0 0 0 0	31
4 2	
5 6 7 8	
8 7 6 5	
0 0 0 0	
1 3 10	
2 4 10	
4 3	
5 6 7 8	
8 7 6 5	
0 0 0 0	
1 3 10	
2 4 10	
2 3 1	
4 3	
5 6 7 8	
30 31 32 33	
0 0 0 0	
1 3 10	
2 4 10	
2 3 1	
4 3	
5 6 7 8	
8 7 6 5	
-1 0 0 1	
1 2 10	
3 4 10	
2 3 1	

Problemsetter: Manzurur Rahman Khan

Special Thanks: Samee Zahur



I Ignore the Blocks



Today is Tilliby's 9th birthday. He has been receiving all kinds of gifts from people like glow-in-the-dark stickers, electronic calculator, a new toothbrush and so forth. However, this funny looking puzzle involving domino tiles caught his attention. The rules were as follows – the puzzle consisted of a rectangular grid of R-by-C cells, and it must be completely filled by 2x1 sized tiles, of which there is sufficient supply. No two tiles may overlap.

To complicate things even more, certain cells of the grid is marked unusable, which must not be covered by any of the tiles. All other cells, however, must be covered by exactly one tile. Now in spite of all these complications, this was an easy exercise for Tilliby. However, when it came to counting the number of ways this can be solved, even his new and shiny calculator could not help him. Can you?

Input

The input consists of at most 100 test cases. Each test case starts with a line containing 3 integers, **R**, **C** and **N**. This line will be followed by **N** other lines, each containing two integers, **r_i** and **c_i**, where **r_i** is the row position of the **ith** unusable cell and **c_i** is the column position. All these input integers will be within the following ranges: $1 \leq R \leq 4$, $1 \leq C \leq 10000000$, $0 \leq N \leq 100$, $0 \leq r_i < R$, $0 \leq c_i < C$.

The last test case will be followed by a single line containing three 0 (zeroes) which should not be processed.

Output

The output for each test case should consist of one single line of the form “Case c: x”, where c is the serial number of the test case starting from 1, and x is the number of ways the specified tiling can be performed. Since this number can be very large, output its value in mod 10000007.

Sample Input	Sample Output
2 2 0	Case 1: 2
2 4 0	Case 2: 5
2 4 2	Case 3: 1
0 0	
0 3	
0 0 0	

Problem setter: Samee Zahur Special Thanks: Md. Arifuzzaman Arif

12669 Blogger language

Benjamin's granddaughter Brenda has a blog where she posts articles about school, friends and other life issues. Intrigued by her opinions, Benjamin tried to read it, but very soon he realized it was too hard to read because of Brenda's writing quirks.

Brenda writes without spaces or punctuation marks, and moreover, she uses lower and uppercase letters in a liberal and strange way. For example, one of her posts is "PrOgRAMmINGiSgrEAt". Benjamin has trouble noticing the words "programming", "is" and "great" when they are written in this way.

To improve his understanding Benjamin decided to do the following: he will first choose a particular string T and a blog post he is interested in; then he will select a contiguous substring of the post and search for T within the substring, in a case-insensitive way; for each occurrence of T within the substring, he will calculate the number of case mismatches, and finally he will obtain the maximum among all these values. For example, if Benjamin chooses "GR" as T and then selects the substring "PrOgRAM", he would find a single occurrence "gR" for which the number of case mismatches is 1. For the same substring, if "r" was chosen as T , he would have found two occurrences, "r" with 0 mismatches and "R" with 1 mismatch, so the maximum number of mismatches would be 1.

To complicate things further, Brenda included in the blog a script that, after operating with a substring selection, flips the case of all the selected letters. This means that after selecting "PrOgRAM" and proceeding as explained above, the sample post would read "pRoGrammINGiSgrEAt". If Benjamin selects "ammINGi" as a second substring, after calculating his result the post would be left as "pRoGrAMminGISgrEAt", accumulating both flips.

You will be given the string T and the original text of the blog post chosen by Benjamin. You will also be given a list of substring selections Benjamin made, in the order he made them. You need to calculate, for each selection, the maximum number of case mismatches of the occurrences of T in the selected part, considering all the case flips made by previous selections. Notice that the flipping of the case occurs after calculating the result for each selection.

Input

The input file contains several test cases, each of them as described below.

The first line contains an integer N ($1 \leq N \leq 10^5$) and a non-empty string T of at most 5 letters, representing respectively the number of substring selections and the string to search for. The second line contains a non-empty string P of at most 10^5 letters, indicating the original text of the blog post. Positions of the post are numbered with consecutive integers from left to right, being 1 the leftmost position and $|P|$ the rightmost position. Each of the next N lines describes a substring selection with two integers L and R ($1 \leq L \leq R \leq |P|$) indicating that the substring starts at position L and ends at position R , inclusive.

Output

For each test case, output N lines, each of them containing an integer. In the i -th line write the maximum number of case mismatches of the occurrences of T in the i -th substring selection, considering all the case flips made by previous selections; if no such occurrence exists write the value '-1'.

Sample Input

```
3 gR
PrOgRAMmINGiSgrEAt
```

```
1 7
4 18
6 14
9 abCAB
aBcAbCAbaBCAb
1 13
1 13
4 8
5 11
3 11
4 10
1 13
8 8
1 13
```

Sample Output

```
0
2
-1
2
4
1
-1
0
5
2
-1
4
```