



**Federal University of Pernambuco**  
Center of Informatics



# UFPE Newcomers Finals 2013

## Programming Marathon

February 8th, 2014

### Problemset

This set contains 11 problems; pages are numbered from 1 to 17, including this front page.

### General Information

#### A) Regarding input

- 1) Your program should read the input from the standard input (*stdin*).
- 2) The input comprises many test cases, each one is described in a number of lines that depends on the problem. The first line of each problem is always the number of test cases.
- 3) When the lines of input contains many values, each one is separated by a single space. There's no other space in the input.
- 4) Each input line, including the last, contains the end-of-line character.

#### B) Regarding output

- 1) Your program should write the output to the standard output (*stdout*).
- 2) When the lines of output contains many values, each one is separated by a single space. There's no other space in the output.
- 3) Each output line, including the last, should contain the end-of-line character.

## Problem A

# Algorithm of Phil

Phil is learning a new algorithm which wasn't taught in his algorithms classes. However, he is not sure whether he implemented it the right way, so he would really appreciate if you could implement it so that he can compare the outputs.

The algorithm starts with a binary string  $A$  and an empty string  $S$ . The algorithm consists of multiple steps. In each step,  $A$  and  $S$  are modified as follows:

- If the number of bits in  $A$  is odd, then the middle bit of  $A$  is added to the end of  $S$  and removed from  $A$ .
- If the number of bits in  $A$  is even, then both middle bits of  $A$  are compared. The bigger one (anyone in case of a tie) is added to the end of  $S$  and removed from  $A$ .
- If after some step the string  $A$  gets empty, the algorithm terminates. The algorithm's return is the decimal representation of the number represented by  $S$ .

*A bit  $a$  is bigger than a bit  $b$  if  $a$  is 1 and  $b$  is 0.*

### Input

The first line contains  $T$  ( $T \leq 500$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case consists of one line containing a binary string  $A$  ( $1 \leq |A| \leq 10^5$ ), representing the algorithm's input.

### Output

For each case print a line containing "Case #X: Y", where  $X$  is the case number, starting at 1, and  $Y$  is the algorithm's return for the given input modulo  $1000000007$  ( $10^9 + 7$ ).

Sample Input	Sample Output
3	Case #1: 0
00000	Case #2: 106
0101101	Case #3: 2
100	

*Author: Duhan Caraciolo*

## Problem B

# Cheap B-Subsequence

Some time ago, Dejan Stojanovic, a Serbian poet, said: "Words rich in meaning can be cheap in sound effects." Is it true? A String Processing professor at UFPE wants to test this quote with strings. For that, he defined what he calls a "cheap B-subsequence". A cheap B-subsequence, according to his definition, is a subsequence of size B, of a string S ( $B \leq |S|$ ), that has the lowest associated cost. To define the cost of a string, the professor determined a series of rules to each letter of the alphabet. The alphabet that he used contains only lowercase letters. The rule of a letter is defined as a set of pairs  $(P_i, C_i)$ , which indicates that if this letter appears in a position X on the subsequence, where X is a multiple of  $P_i$ , then the cost of  $(X/P_i)*C_i$  will be added to the total cost of this subsequence. Let's show an example. Suppose we have the following rules:

$$[a] = \{(2,3), (4,10)\}$$

$$[b] = \{(1,4), (7,50)\}$$

$$[c] = \{(1,2), (4,20)\}$$

$$[d..z] = \{\} \text{ // there are no rules for the characters 'd' to 'z'}$$

Suppose we have the string *abaabcbc*, and  $B = 4$ . If we choose the subsequence *aabc* (*abaabcbc*), we would do the following procedure to calculate the associated cost:

1. The first letter of the sequence is an 'a', and the position 1 is neither multiple of 2 or 4, so the cost is 0;
2. The second letter of the sequence is another 'a', and the position 2 is a multiple of 2, so we'll add the cost of  $(\frac{2}{2}) * 3 = 3$ ;
3. The third letter of the sequence is a 'b', and the position 3 is multiple of 1, so we will add the cost of  $(\frac{3}{1}) * 4 = 12$ ;
4. The last letter of the sequence is a 'c', and the position 4 is a multiple of 1 and 4, so we will add the cost of  $(\frac{4}{1}) * 2 + (\frac{4}{4}) * 20 = 28$ .

The total associated cost to this subsequence is 43, which is not the lowest cost, since we could have chosen *aaab* (*abaabcbc*) and obtained an associated cost of 19 - this is indeed the cost of the cheap B-subsequence. Given the string S and the integer B, and the rules of the alphabet, your task is to create a program that tells the professor the cost of the cheap B-subsequence.

### Input

The first line contains T ( $T \leq 100$ ) – the number of test cases, after this line T test cases follows. The first line of a test case contains a string S of lowercase letters and an integer B ( $1 \leq B \leq |S| \leq 100$ ). Each of the next 26 lines describe the rule of each letter. The first of the 26 lines corresponds to the rule of the letter 'a'; the following line corresponds to the rule of the letter 'b'; the last of the 26 lines corresponds to the rule of the letter 'z'. Each line containing a rule is described in the following way: Q P<sub>1</sub> C<sub>1</sub> P<sub>2</sub> C<sub>2</sub> ... P<sub>Q</sub> C<sub>Q</sub> ( $1 \leq Q \leq 10$ ;  $1 \leq p_i \leq |S|$ ;  $1 \leq c_i \leq 50$ ), where Q is the amount of pairs associated to this rule, and is followed by the pairs themselves.

### 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 cost of the cheap B-subsequence.

<b>Sample Input</b>	<b>Sample Output</b>
2 abcd 1 1 1 20 1 1 15 1 1 8 1 1 30 1 1 2 0 ( <b>21 lines</b> ) abaabcbc 4 2 2 3 4 10 2 1 4 7 50 2 1 2 4 20 0 ( <b>23 lines</b> )	Case #1: 8 Case #2: 19

Author: Gustavo Stor

## Problem C

# Cryptography of the Floating Key

Some students of Computer Science at UFPE decided to create a project based on the Peer-to-Peer model to transfer data. They also decided to implement a unique method of cryptography based on symmetric-keys algorithms so that any messages could be encrypted and decrypted using the same key. They called this method “cryptography of the floating key”. It receives as input a string  $P$  of lowercase letters and a key  $(N,M)$ , and returns the encrypted message  $Q$ . We'll now describe the steps to encrypt a string  $P$  with a key  $(N,M)$ .

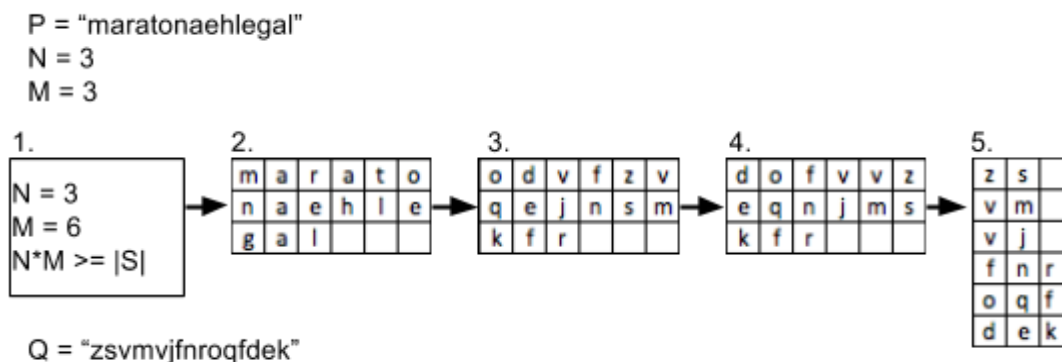
1. If  $N*M \geq |P|$ , go to step 2. Otherwise, if  $N$  and  $M$  are different, take the smallest of them and increment it with the value of the largest of them; if they are equal, double the value of  $M$ . Return to step 1.

2. Create a  $N \times M$  matrix and arrange the string  $P$  in this matrix: the first line of the matrix should be occupied by the first  $M$  characters of  $P$ , the second line of the matrix should be occupied by the following  $M$  characters of  $P$ , and so on. There'll probably be some empty cells on the matrix by the end, but you should not worry about it.

3. Sweep each element of the matrix that is not an empty cell and check its position. The rows and columns are indexed starting from 1. If an element is on the position  $(i,j)$  of the matrix, then make a shift of  $i+j$  on the character. That is, if there's a character 'a' on the position  $(1,1)$  of the matrix, it turns to 'c'; if there's a character 'z' on the position  $(1,3)$  of the matrix, it turns to 'd'.

4. From the first row and first column, swap adjacent elements of the matrix in a zigzagging way. That is, given that you are in position  $(1,1)$ , you'll swap the element at  $(1,1)$  with the element at  $(1,2)$ . Then, go to the right. Now, you are in position  $(1,2)$ , but the element at  $(1,2)$  has already been swapped, so you don't swap it and keep going right. When you get to position  $(1,M)$ , if it has been swapped, just go down to position  $(2,M)$ . If it hasn't, swap it with the element  $(2,M)$  and go down. Now that you are at  $(2,M)$ , you'll start going left, applying the same algorithm, and when you get to  $(2,1)$ , you'll go down and start going right. The line of the matrix with the last part of  $P$  should be left intact, so you can't swap any elements in that line. If you come to a situation where you have to swap an element with the element below it, but the line below is the line containing the last part of the string  $P$ , then you stop immediately.

5. Now you should rotate the matrix  $90^\circ$  degrees to the left, and the string should be read from left to right, top to bottom. Empty spaces of the matrix should be ignored. As result, you'll obtain the encrypted string  $Q$ .



Somehow, the piece of code responsible of decrypting the messages was infected by a jealous virus, and they called you to create a new program. You must solve the following problem: given the encrypted message  $Q$  and the key  $(N,M)$ , decrypt the message and return the initial string  $P$ .

**Input**

The first line contains  $T$  ( $T \leq 10^4$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case is arranged in a line containing a string  $Q$  ( $1 \leq |Q| \leq 100$ ) and two integers  $N$  and  $M$  ( $1 \leq N, M \leq 100$ ) – the encrypted message and the two values that make up the key, correspondingly.

**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 initial string  $P$ .

<b>Sample Input</b>	<b>Sample Output</b>
3 vnxjbxylndjhzq 1 1 zsvmvjfnroqfdek 3 3 uhyeobnqcmayftjvttrplnraykamqw 50 100	Case #1: letshavesomefun Case #2: maratonaehlegal Case #3: universidadefederaldepernambuco

*Author: Gustavo Stor*

## Problem D

# Dudu, the Possum

Dudu is a very starving possum. He currently stands in the first shelf of a fridge. This fridge is composed of  $N$  shelves, and each shelf has a number  $Q_i$  ( $1 \leq i \leq N$ ) of food. The top shelf, where Dudu is, is identified by the number 1, and the lowest is identified by number  $N$ . Dudu doesn't eat more than one food in the same shelf, because he doesn't want to get noticed. Furthermore, Dudu is very fat and cannot climb the wall of the fridge to a shelf above — nobody knows how did he end up in the first shelf. Dudu is also afraid of height, so he is only able to climb down at most  $K$  shelves at a time (if he is at shelf  $i$ , he is only able to reach shelves  $i+1, i+2, \dots, i+K$ ). There is a chance  $P_j$  that he chooses to get down from a shelf  $i$  to a shelf  $i+j$  ( $1 \leq j \leq K$ ). If he tries to go down a number of shelves that makes him get past the lowest shelf, he gets out of the fridge — he will always get out of the fridge eventually, because someone left the door open. Each food of shelf  $i$  has a number of calories  $C_{i,j}$  that is absorbed by Dudu in case he eats it, and a probability  $X_{i,j}$  that it is chosen by Dudu, for  $j$  from 1 to  $Q_i$ . Dudu starts his journey at shelf 1 and, when he is in a shelf, he will always choose a food to eat and then will go to another shelf. What is the expected number of calories that Dudu will absorb by the time he gets out of the fridge?

### Input

The first line contains  $T$  ( $T \leq 100$ ) – the number of test cases, after this line  $T$  test cases follows. The first line of a test case contains two integers,  $N$  and  $K$  ( $1 \leq N \leq 500$ ;  $1 \leq K \leq 10$ ) – the number of shelves in the fridge and the maximum number of shelves Dudu can climb down at a time, correspondingly. The second line of a test case contains  $K$  real numbers  $P_j$ , where  $P_j$  is the probability that Dudu goes down  $j$  shelves, for  $j$  from 1 to  $K$  ( $0 \leq P_j \leq 1$ ;  $\sum_{j=1}^K P_j = 1$ ). Each of the next  $N$  lines of a test case describes a shelf (from the shelf 1 to shelf  $N$ ). Each line starts with an integer  $Q_i$  ( $1 \leq Q_i \leq 20$ ), which is the amount of food existent is in this shelf.  $Q_i$  pair follows, each pair containing 2 real numbers  $C_{i,j}$  and  $X_{i,j}$  ( $0 \leq C_{i,j} \leq 100$ ;  $0 \leq X_{i,j} \leq 1$ ;  $\sum_{j=1}^{Q_i} X_{i,j} = 1$ ).

### 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 expected number of calories that Dudu will absorb by the time he gets out of the fridge.  $Y$  should be rounded up to 6 digits after the decimal point.

Sample Input	Sample Output
2	Case #1: 90.000000
2 1	Case #2: 44.929950
1.0	
2 50 0.5 100 0.5	
2 10 0.5 20 0.5	
5 2	
0.3 0.7	
5 10 0.2 20 0.3 5 0.1 25 0.35 2 0.05	
2 20 0.4 40 0.6	

1 4 1.0 3 30 0.8 3 0.1 4 0.1 10 1 0.1 2 0.1 3 0.1 4 0.1 5 0.1 6 0.1 7 0.1 8 0.1 9 0.1 10 0.1	
---	--

*Author: Lucas Lima & Gustavo Stor*



## Problem E

# Hnelpig Arnde

Did you konw taht, if we mses up all the letetrs of a wrod, ecxlundig the first and lsat oens, the wrod rmeanis prefcelty unedrsantdbale?

Yep. Since researchers at UFPE found that out, it became common in the internet to write everything messed up. LifeInvader posts, chatting rooms, blogs, professional e-mails, academic works, every kind of text became victim of this modern trend. Everybody got used to write with messed letters and understand what is written.

Except Andre. Andre is kind of a dumb guy. Andre isn't inside all the internet trends. Andre has no clue. Andre can't read messed texts, and because of that he has problems to study, communicate at League of Legends and even chat with the chicks.

Fortunately, Andre has a programmer friend to help him with that problem: you! He asked you to create a program to fix the messed texts using a set of words. It's up to you to help him live his life normally. Help him and he will call a friend to get you an ACC!

### Input

The first line contains  $T$  ( $T \leq 100$ ) – the number of test cases, after this line  $T$  test cases follows. The first line of a test case contains two integers  $N$  ( $N \leq 100$ ) and  $M$  ( $M \leq 20$ ) – the number of words in the set and the number of lines of text Andre wants to fix, correspondingly. The next line of a test case will contain  $N$  space-separated words – the words in the set. The next  $M$  lines of a test case will contain the text lines to be fixed, with at least one and at most 50 messed up words, and every word will be separated by a single space. Every word in the input will have at most 40 letters and will contain only lowercase letters. It's guaranteed that every word of the input has exactly one corresponding word in the set.

### Output

For each test case print a line containing "Case #X:", where  $X$  is the case number, starting at 1. Then, for every line of text, print it with the words fixed according to the set.

Sample Input	Sample Output
<pre>2 9 2 andre copy exam free gimme lemme plox stuff yo ardne lmeme cpoy yo eaxm polx adnre gmime fere sfutf polx 4 1 abacabadabacaba abacaba aba a a aba acbbaaa adccbbbbaaaaaa</pre>	<pre>Case #1: andre lemme copy yo exam plox andre gimme free stuff plox Case #2: a aba abacaba abacabadabacaba</pre>

Author: Mário Henrique

## Problem F

# Fat and Orial

Fat is a very popular kid, but unfortunately his grades are not directly proportional to his amount of friends. Orial, one of Fat's friends, was determined to help Fat and decided to call you to make a program. Currently, the total average grade of Fat is  $N$ , and he has already attended  $A$  disciplines. Fat wishes to achieve the total average grade of  $M$  by the time he has attended  $B$  more disciplines. Help Orial to make a program that helps Fat estimate the average grade he should score on the next  $B$  disciplines so that he achieves the total average grade of  $M$ . The grading system is such that it's impossible to score a grade lower than 0 or bigger than 10.

### Input

The first line contains  $T$  ( $T \leq 10^5$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case is arranged in a line containing the numbers  $N$ ,  $M$ ,  $A$  and  $B$  ( $0 \leq N, M \leq 10$ ;  $1 \leq A, B \leq 100$ ) in a single line – Fat's current total average grade, desired total average grade, amount of disciplines he has attended and amount of disciplines he will attend, correspondingly.  $N$  and  $M$  are real numbers, and  $A$  and  $B$  are integers.

### Output

For each test case print a line containing "Case #X: Y", where  $X$  is the case number, starting at 1, and  $Y$  is either a floating point number, which is the average grade that Fat should score from now on to obtain the desired total average grade, or "Impossible" if it's impossible to obtain such an average grade. In the case where it's possible to achieve the goal,  $Y$  should be rounded up to 2 digits after the decimal point. The input will be in a way that errors up to  $10^{-3}$  will still give the correct answer.

Sample Input	Sample Output
3 7 8 1 1 7 9 5 2 9.05 9.20 28 20	Case #1: 9.00 Case #2: Impossible Case #3: 9.41

Author: Gustavo Stor

## Problem G

# 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 LifelInvader (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 LifelInvader, 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 LifelInvader, 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 LifelInvader (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 LifelInvader 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	Sample Output
3	Case #1: 0
3 2 1	Case #2: 4
1 2	Case #3: -1
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	

Author: Mário Henrique

## Problem H

# The Sightseeing Tour

Skyrk is now living in an old town, full of ancient tales and wonders to behold. No surprise this town became one of the biggest tourist attractions in the world. Unfortunately, it is rather difficult to move around the town, since it is very old and surrounded by mountains, its streets are narrow, full of turns and ups and downs. But this was rather fortunate for Skyrk. He thought a lot of tourists would rather see the touristic places looking from the top of the mountains instead of adventuring in the old town. He decided to set sightseeing binoculars on some spots of the biggest mountain near the town, and of course, charge for their use.

The town has  $N$  touristic places, looking from the mountain, they seem to line up from left to right. Skyrk set up  $M$  sightseeing binoculars, each can cover the view from touristic place  $A$  up to  $B$  and cost  $C$  coins to use. A group of tourists arrived to do a sightseeing tour, and they want to see all touristic places paying the minimum price. Skyrk knows in advance the price each tourist is willing to pay for the tour. In order to maximize his profit, he will tell the tourist that only a subset of binoculars are working, so that when the tourist choose the ones he wants, he ends up paying more.

### Input

The first line contains  $T$  ( $T \leq 100$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case starts with a line with three integers  $N, M, K$  ( $1 \leq N \leq 10^9$ ;  $1 \leq M \leq 30$ ;  $1 \leq K \leq 10^4$ ) – the number of touristic places, sightseeing binoculars and groups of tourists, respectively. Then there will be  $M$  lines with three integers  $A, B, C$  ( $1 \leq A \leq B \leq N$ ;  $1 \leq C \leq 10$ ) – the first and last touristic place the binocular can see and its price, respectively. Then there are  $K$  lines with two integers  $X, Y$  ( $1 \leq X \leq Y \leq 10^9$ ) – there will be a group of tourists where the first is willing to pay  $X$ , the second  $X+1$ , the third  $X+2$  and so on until the last one who is willing to pay  $Y$ . Every touristic place can be seen by at least one binocular.

### 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 maximum profit Skyrk can get.

Sample Input	Sample Output
2	Case #1: 11
3 4 2	Case #2: 135
1 1 4	
1 2 1	
2 3 2	
3 3 1	
1 3	
10 10	
5 6 2	
1 2 5	
2 2 1	
3 5 10	
3 4 10	

4 5 10 5 5 5 12 17 23 26	
-----------------------------------	--

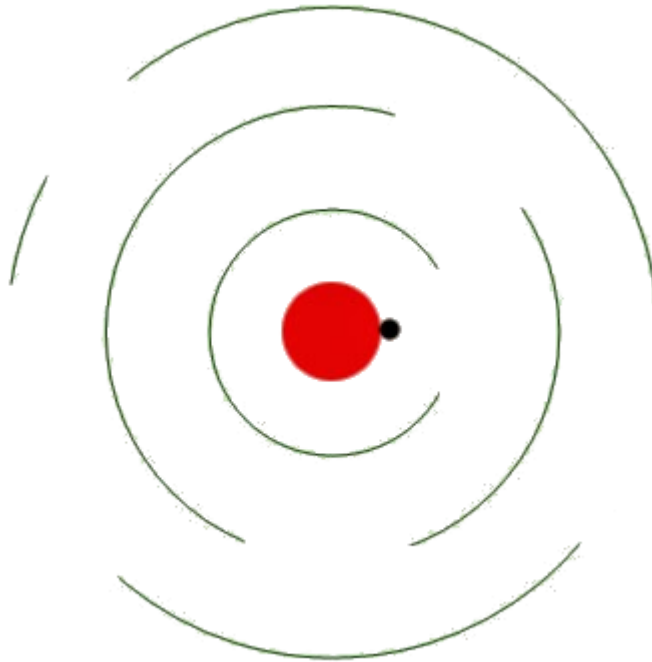
Consider the first test case, there are 3 touristic places, 4 binoculars and 2 groups of tourists. The first group of tourists go as follows: The first tourist can pay at most 1 coin. There is no subset Skyrk can choose that satisfy the tourist, so the tourist don't take the tour and pays nothing. The second tourist can pay at most 2 coins, the only choice Skyrk has is to put the binoculars 2 and 4 on the subset, the tourist then chooses binoculars 2 and 4 and pays 2 coins. The third tourist can pay at most 3 coins. Skyrk can put binoculars 1, 2 and 3 on the subset, the tourist then chooses binoculars 2 and 3 and pays 3 coins. The second group of tourists go as follows: The first tourist can pay at most 10 coins. Skyrk can put binoculars 1 and 3 on the subset, the tourist then chooses binoculars 1 and 3 and pays 6 coins. The maximum profit Skyrk can get is  $2+3+6 = 11$  coins.

Author: Davi Duarte Pinheiro

## Problem I

# Super Circumference

Skyrk has developed a game called Super Circumference. Its simplicity is only equaled by its monumental difficulty. The goal of Super Circumference is to control a point which circles around a central circumference attempting to avoid contact with incoming circular walls.



The point circles around at the speed of one full revolution per second. A level has several sets of incoming circular walls which the point has to avoid contact. An incoming wall can be regarded as a circumference sector. The set of walls come at the speed of one set every  $P$  seconds. The player wins the game if he successfully evade all walls.

The red circle is the center which the black point circles around. The incoming green walls approach the red circle. A brand new level has been created with  $N$  sets of walls, to adjust its difficulty properly, you are to find the minimum possible  $P$  that still makes the level possible to complete.

### Input

The first line contains  $T$  ( $T \leq 100$ ) – the number of test cases, after this line  $T$  test cases follows. The first line of a test case contains one integer  $N$  ( $1 \leq N \leq 10^4$ ) – the number of sets of walls. Next  $N$  lines contain the description of the set of walls in the following form: First the number  $K$  ( $0 \leq K \leq 10$ ) – the number of walls this set has. Next, there'll be  $K$  pairs of floating point numbers  $X$  and  $Y$  ( $0 \leq X, Y < 2\pi$ ) – the wall starts at  $X$  and extends along the circumference in clockwise direction until  $Y$ . Walls of the same set do not intersect. The circumference sector covered by a wall will not be empty and will not be a full circumference. When the game starts the first wall will hit the center after  $P$  seconds and the black point can start at any position. The sets of walls are given in the order they approach the center.

**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 possible P that makes the level possible to complete. Y should be rounded up to 4 digits after the decimal point. The input will be in a way that errors up to  $10^{-5}$  will still give the correct answer.

<b>Sample Input</b>	<b>Sample Output</b>
2 3 1 1 5.28 2 1.5 4.28 5 1 3 0 2 2.5 3 3.8 4.5 2 2 0 1 2 3 2 0.9 2.1 2.9 0.1	Case #1: 0.0446 Case #2: 0.0159

Author: Davi Duarte Pinheiro

## Problem J

# Squares Game

What could be Ana and Bob doing while the world is being destroyed? Playing, of course!

It's a simple game, in a rectangular board with  $N$  rows and  $M$  columns ( $N \times M$  cells). Ana starts playing: she can pick any position in the board where there's a square of  $2 \times 2$  unmarked cells and mark them. Then, it's Bob's turn: he can pick any unmarked cell and mark it. If a player has no move, he passes his turn, until there's no unmarked cells. Who marks the largest number of cells wins the game.

Ana and Bob are optimal players, and they always play to win (or draw if they cannot win).



Possible sequence of a game between Ana and Bob, where Ana marks red and Bob marks blue. Note that, by the end of this sequence of turns, Ana won't be able to mark further cells and will always pass her turn to Bob.

### Input

The first line contains  $T$  ( $T \leq 10^4$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case is arranged in a line containing two integers  $N$  and  $M$  ( $1 \leq N, M \leq 100$ ) – the number of rows and the number of columns of the board, correspondingly.

### Output

For each test case print a line containing "Case #X: Y", where  $X$  is the case number, starting at 1, and  $Y$  is "Ana" (without the quotes) if she can win independently of how good Bob is; "Bob" (without the quotes) if he can win independently of how good Ana is; or "Draw" (without the quotes), otherwise.

Sample Input	Sample Output
3	Case #1: Ana
2 3	Case #2: Draw
2 4	Case #3: Bob
3 3	

Author: Luiz Afonso



## Problem K

# Skyrk's Bar

Skyrk has opened a bar which sells a new kind of beer. Its color is blue, and unlikely most beers, the more you drink, you don't get drunk, you get smarter! Of course Skyrk's bar was a widespread success and people all over the world came to taste the vintage.

Too much beer makes people go to the bathroom very often, and the bathroom quickly became too crowded. This was especially true for the men's bathroom, since men have a widely known custom of only using an urinal if it has  $K$  empty urinals between him and any other man. This became a problem to Skyrk because despite having several urinals, they couldn't be all used at the same time.

When a man goes to the bathroom, he chooses at random an urinal that is available given men's picky custom. If there is no spot for him, he gets frustrated and leaves the bar. Skyrk thinks that if the bathroom had  $N$  urinals, that would solve the problem.

But to be certain, Skyrk devised an experiment: The bathroom starts empty, and men will pick available spots at random until there is no available spot. No man will leave the bathroom in the meantime. What is the expected number of men that can be using the bathroom at the same time when it becomes full?

### Input

The first line contains  $T$  ( $T \leq 100$ ) – the number of test cases, after this line  $T$  test cases follows. Each test case is arranged in a line containing two integers  $N$  and  $K$  ( $1 \leq K \leq N \leq 10^6$ ) – the number of urinals and the number of needed free urinals between any pair of men, correspondingly.

### 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 expected number of men that can be using the bathroom at the same time.  $Y$  should be rounded up to 4 digits after the decimal point. The input will be in a way that errors up to  $10^{-5}$  will still give the correct answer.

Sample Input	Sample Output
3	Case #1: 1.5000
4 2	Case #2: 2.2857
7 2	Case #3: 2.4133
10 3	

Author: Davi Duarte Pinheiro