IUT 6TH NATIONAL ICT FEST 2014
SEPTEMBER 12TH & 13TH

Organized by
CSE Department, IUT
&
IUT Computer Society

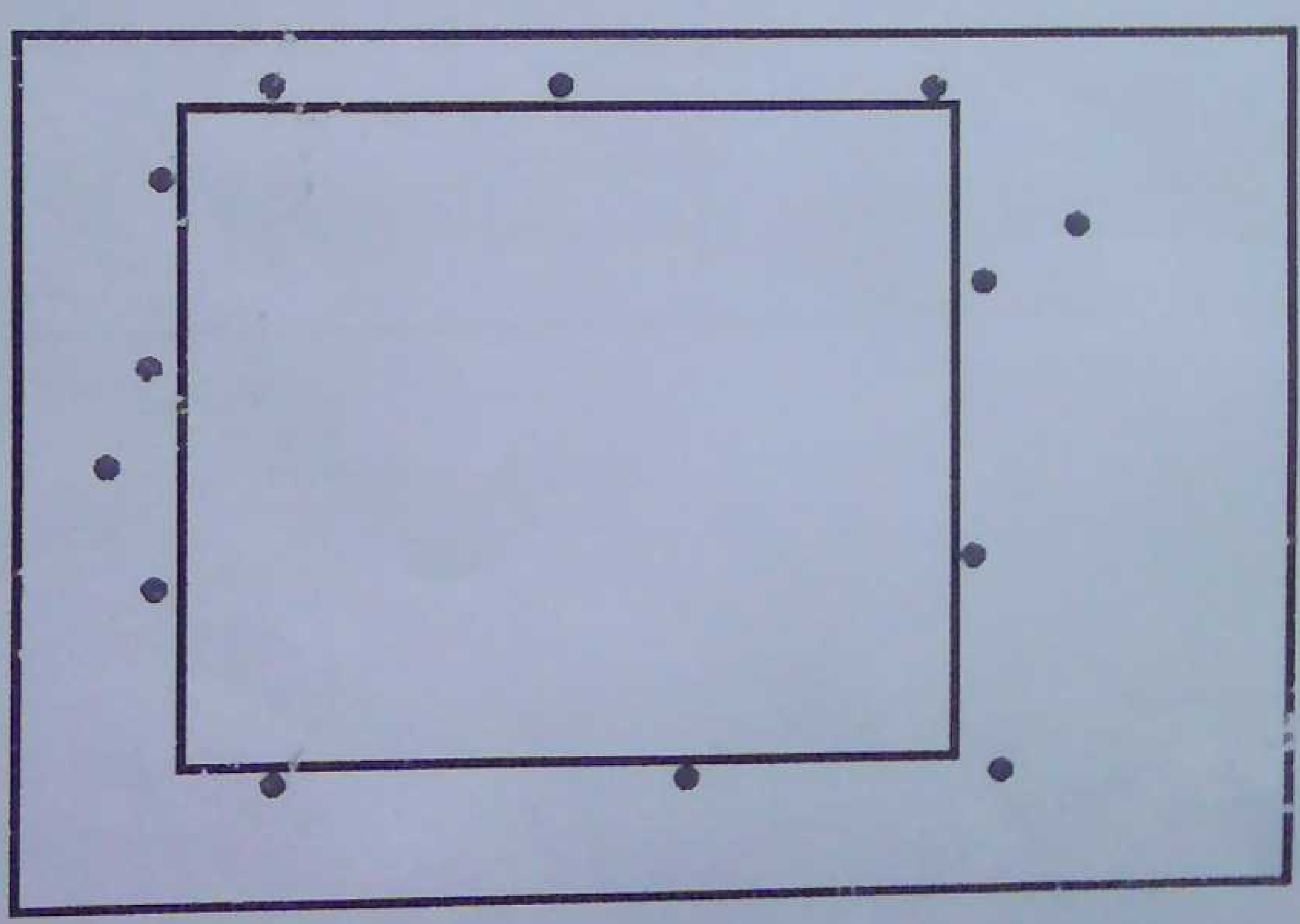
Problem Set of Programming Contest

A   A Football Stadium

Once upon a time, Sand King used to rule the sand country. In the sand country, there was sand everywhere, but there were a very few sources of oxygen. In the sand country, trees were worshipped, as without them sand people would not live. But the Sand King was very much fond of football. So he ordered to make a very BIG football stadium (rectangular in shape) in his kingdom. The news was very alarming to all the citizens of the sand kingdom. As cutting the very few trees would cause death to the living creatures. So they requested the king to not make the football stadium.

The king was so stubborn in his idea that he would not give up making a football stadium. But he was also sympathetic to his citizens. So he has hired you to help him out.

You are given the width and height of the sand kingdom. Fortunately, sand king wants the football stadium to be parallel to the borders of the sand kingdom (he does not like bad orientation of the football stadium). You are also given the positions of the valuable trees. Your task is to maximize the area of the football stadium so that no trees are harmed. You can consider the trees as points. The trees may stand at the boundary of the football stadium, but cannot be strictly inside it. Note that the Kingdom is a rectangle with axis parallel sides and the lower left point of the kingdom is (0, 0) and upper right point is (L, W).



Input

The first line of the input contains a number **T (1≤ T≤ 100)**, denoting the total number of test cases. Next, **T** test cases follow. Each test case begins with two numbers, **L** and **W**

A - 2

($1 \le L,W \le 10000$), denoting the length and width of the sand kingdom respectively. Then a number $N(1 \le N \le 200)$. Followed by, there are **N** integer co-ordinates of trees, x_i, y_i, where ($0 \le x_i \le L$, $0 \le y_i \le W$).

Output

For each input set, first print the case number, followed by the maximum possible area of the football stadium. See the sample input/output for more details.

Sample Input	Output for Sample Input
3 4 8 3 1 2 3 4 3 7 12 10 2 3 6 8 9 5 5 2 0 0 5 5	Case 1: 18 Case 2: 81 Case 3: 25

B Bob the Builder

Bob "the builder" builds integers! He has his own IBM (Integer building machine) to do that. It works like this:

Bob takes an Integer **X**. For example **X** is 13. We know that every positive integer can be written as sum of some unique power of 2. So Bob can write 13 as 1+4+8. Now 13 can have one of these 3 children:

13+1=14

13+4=17

13+8=21

Similarly children of 12 are: 12+4=16

12+8=20

That's because 12 can be written as 4+8.

Now Bob uses IBM to build children from an Integer. Bob takes **X** and using the IBM just once he can do following things:

1. Build just one child from X, let's say the child is Y.
2. Build just one child from Y, let's say the child is Z.
3. Build just one child from Z, let's say the child is Q.

   And so on…

Ok this can go forever, so Bob never builds an integer larger than a limit L. At each step, Bob can select which child to build. So if the limit is L=24 and X=12, Bob can build 20 from 12 and then 24 from 20 just using the IBM once. Also he can build 16 from 12 using the IBM another time. **One integer can be used multiple times to create new child, but Bob never builds same integer more than once.**

3

Bob has a list of N integers in his factory. In that list no integer is direct child or even descendant of another. Bob wants to build all possible children from those integers. But using IBM is very costly, it consumes lots of electricity. So your task is to calculate how many times he must use IBM to do it!

Input

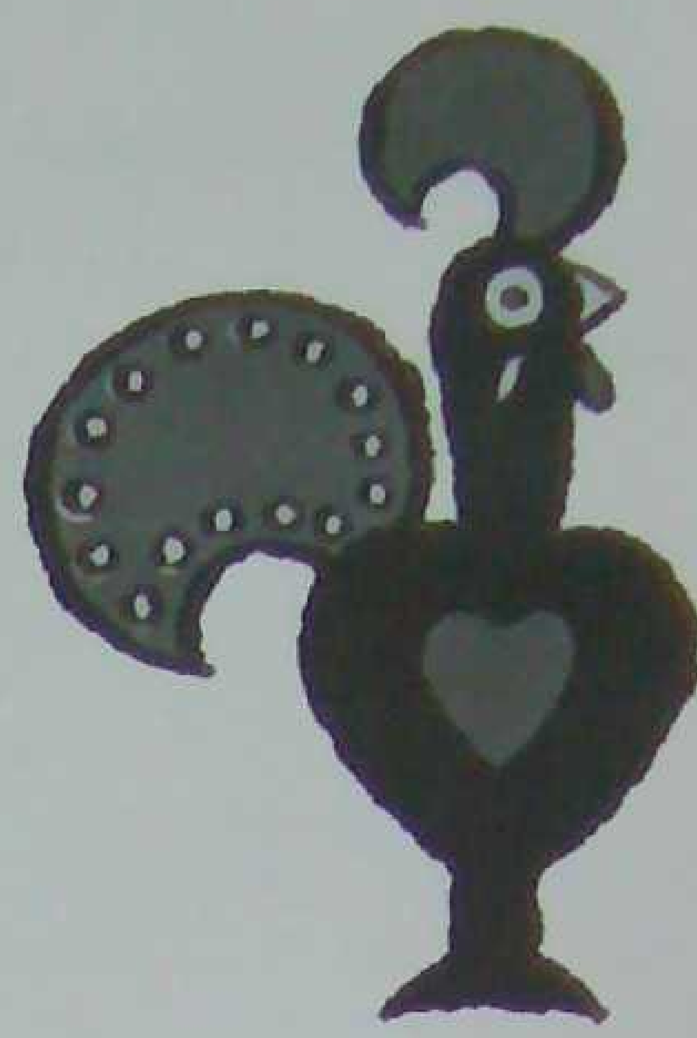
Input starts with an integer, **T** **(T ≤ 100)** denoting the number of test cases. Each case has two lines. First line consist two integers **N** (1≤**N**≤36) and **L** (1≤**L**≤10000). In next line there are **N** space separated positive integers **(≤L)** representing the list Bob has in his factory.

Output

For each case, print the case number and the answer, number of times Bob has to use the IBM. See sample output for exact format.

Sample Input	Output for Sample Input
2 1 36 20 2 40 8 20	Case 1: 2 Case 2: 3

C Chicken Lover



Abir loves to eat. Every time he visits a restaurant he wants to eat a chicken item. But chicken item may not be always available. In each day he visits **m** restaurants consecutively. Each restaurant $(i = 1\ldots m)$ can make $\mathbf{n_i}$ different items (Number of Chicken item is exactly 1). But in a single day each restaurant prepares exactly $\mathbf{k_i}$ items (chosen randomly from $\mathbf{n_i}$ items).

Find expected number of chicken items Abir can eat in a single day.

Input

Input starts with an integer **T** ($\le$ **125**), denoting the number of test cases. Each case starts with a line containing an integer **m** ($1 \le m \le 10000$) which denotes number of visiting restaurants. Then in the following line there will be **m** pair of numbers $\mathbf{n_i}$ and $\mathbf{k_i}$ ($1 \le i \le m$, $1 \le n_i \le 20$, $1 \le k_i \le n_i$).

Output

For each case, print expected number of chicken items Abir can eat in a single line in the format **P/Q,** where P and Q are relatively prime (i.e. no common factor > 1, between **P** and **Q**).

Sample Input	Output for Sample Input
3 1 1 1 2 2 1 2 1 1 3 2	Case 1: 1/1 Case 2: 1/1 Case 3: 2/3

$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times 1 \times 1$ 2 3 5

$\frac{1}{8}$

$x \to n$

$j = n$

2 3

Explanation for Sample Case

In the first case, total no of item is one (one chicken item) and probability of getting 1 chicken item is one. So expected number of chicken item is 1.

In the second case, probability of getting 1 chicken item is ½ and probability of getting 2 chicken items is ¼. So expected no of chicken item is $1 \times \frac{1}{2} + 2 \times \frac{1}{4} = 2/2 = 1/1$.

D Daily Potato

Moshtak is a reporter of daily potato. He likes to play with names. Whenever he writes someone's name in his paper, he changes it to his own will. Once he changed Priyoti's name. Priyoti got very angry. She invites him to play a game of strings. If he loses the game he has to change his name to potato. The rule of the game is given below.

Moshtak will be given a string **S** of lower case alphabets. Priyoti will make some queries. In each query moshtak will be given a character **C** and an integer **X.** He has to tell the number of palindromes which are substrings of **S,** starts and ends with **C** and each contains exactly **X** occurrences of **C.**

Moshtak does not want to be a potato. Please help him to win the game.

Input

Input starts with **T**, the number of test cases to follow. First line of each test case is an integer **N**. In the next line will contain a string **S,** consist of **N** lower case characters. In the next line an integer **Q** is given which is the number of query. In the next line a string **QS** consisting of **Q** characters is given i-th character of **QS** will be the **C** in i-th query.

In the next line **Q** integers separated by space is given. i-th integer will be the **X** in the i-th query.

All the integers in input will fit into 32 bit signed integer.

All the strings in input will contain only lowercase alphabets and

$1 \le T \le 20$, $1 \le |S| \le 100000$, $1 \le Q \le 100000$

Output

For each test case, print the test case number as given in the sample output. Then for each query output the number of palindromes which are substrings of **S** and starts and ends with **C** and each contains exactly **X** occurrences of **C.**

Sample Input	Output for Sample Input
1 8 abccbaab 6 abcabc 2 2 2 1 1 3	Case 1: 2 2 1 3 3 0

E Extreme Terror

"Kana Shamsu" is a frightening local terror. There are **N** business places in his area. He (Kana Shamsu) claims extortion to the business owners to continue their business, otherwise they will suffer. Shamsu is a dangerous man. There is a rumor that Shamsu can kill people for money. The total business society is feeling unsafe. It comes as an unexpected disaster.

But Shamsu has his own boss too. He works under a very powerful godfather named "Vuri Kamal". Kamal is a very busy person, as he needs to manage lots of local terror like Shamsu. So he (Kamal) doesn't want to know how much money Shamsu can receive from the business owner. He claims Shamsu to give him some specific amount of money X_i for those business places (where **i = 0** to **N-1**). On the other hand, business owners want to make an agreement with Shamsu for the donation amount Y_i for each business (where **i = 0** to **N-1**).

So it comes to a situation that, for each business place **i** ($0 \le i < N$) Shamsu gets Y_i from the business owner and needs to give the Kamal X_i. Meanwhile Shamsu realizes that, for some business places, his godfather Mr. Vuri Kamal claims more than he can manage from that place. To some extent, Vuri Kamal gives Shamsu a chance to deny collecting money from at most **K** places. So now Shamsu needs to find a process to maximize his profit he can gain from these **N** business places.

Input

There will be **T** ($T \le 100$) test cases. For each test case, there will be three lines of input. In the first line you will be given **N** ($1 \le N < 10^6$) and **K** ($0 \le K \le N$). The second line contains **N** space separated integers X_i ($0 \le X_i < 10^8$), denoting the amount of money Vuri Kamal claims from Shamsu for each business place. The next line will contain another **N** space separated integers Y_i ($0 \le Y_i < 10^8$) denoting Shamsu can get from the business owner.

Output

In the output, you will print the maximum profit Shamsu can earn for the given input. If it is not possible to earn positive profit (total profit ≤ 0) for a given input, you will print "**No Profit**" without the quote. For further instruction, follow the sample input output given bellow.

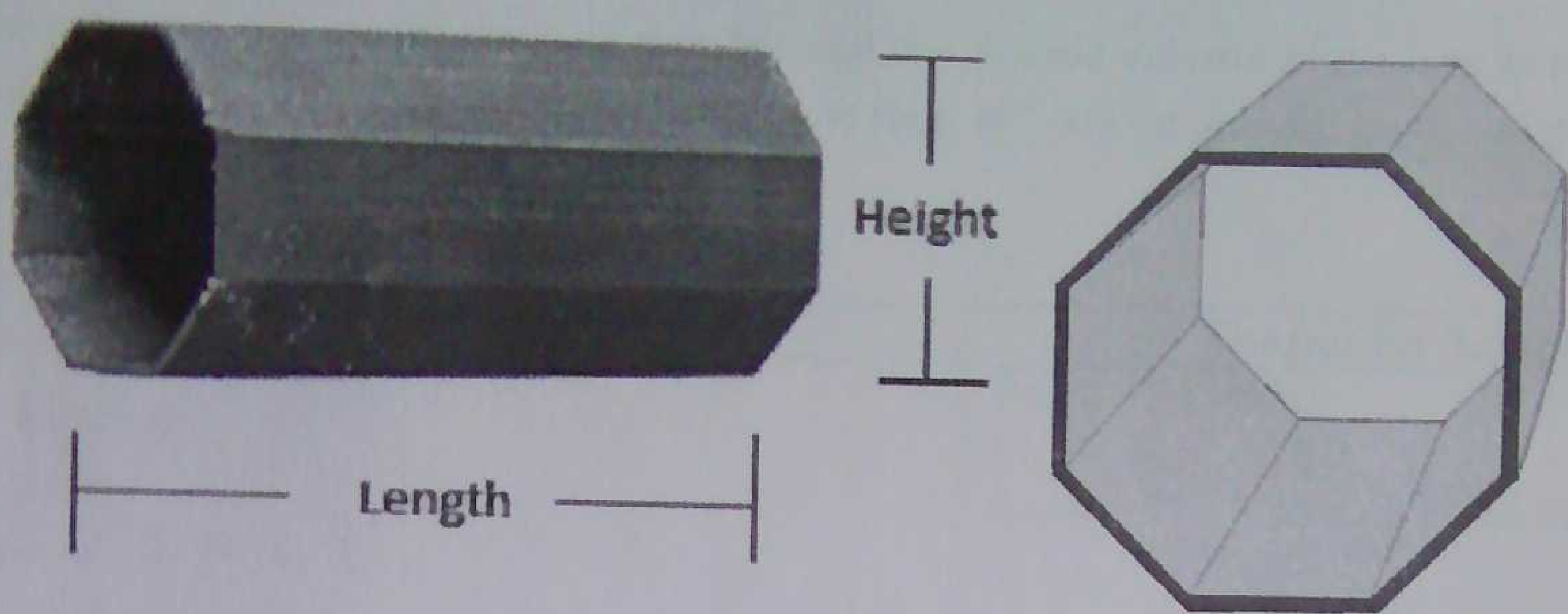
Sample Input	Output for Sample Input

`2`
`5 2`
`4 5 6 7 8`
`5 6 7 8 9`
`5 0`
`7 6 7 8 7`
`4 5 6 7 8`
	`Case 1: 5`
	`Case 2: No Profit`

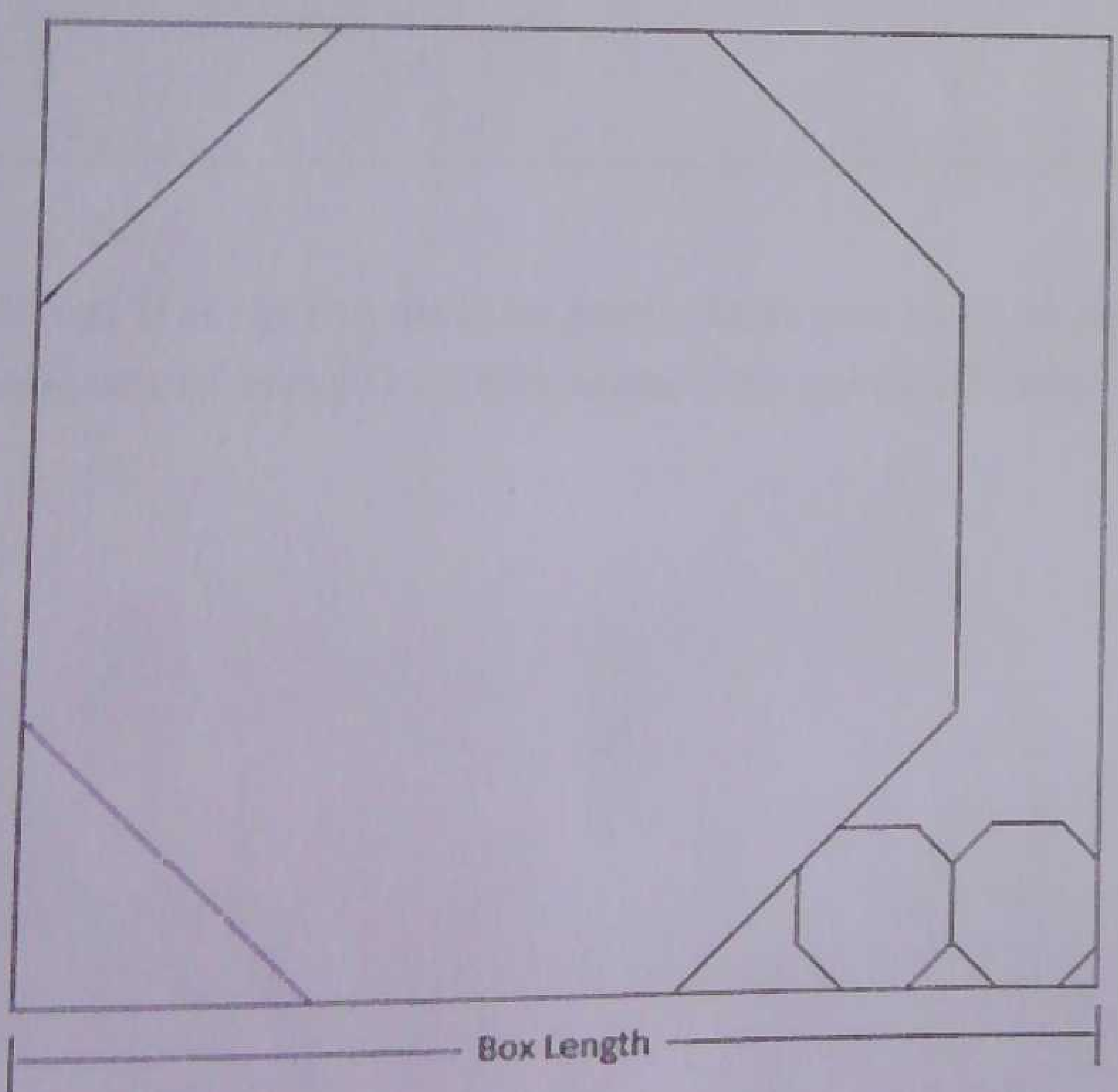
F

Fitting Pipes Again

We are making a box to pack some *regular octagonal* pipes. The box is *8* feet wide, and the length of each pipe is also *8* feet. But they have different cross sectional heights. A regular octagonal pipe has equal cross sectional height and width. If you are unfamiliar with octagonal pipes, see pictures below.



These pipes will be placed in the box such that, their lengths align with the width of the box and one of the eight sides must touch the box's floor completely, which means, we cannot stack them one above the other. The next picture will show you a particular packing example, where three pipes have been packed in the box.



If you are given the cross sectional heights of pipes we wish to pack in the box, can you calculate the minimum possible volume for the box? You can change the order of pipes but you have to follow the rules for placing them in the box.

You can assume, the number of pipes will be at most 8, height of each pipe is no greater than 100,000 feet and there can be at most 100 test cases.

Input

First line of the input will be an integer T ($1 \le T \le 100$) which indicates the number of test cases to follow. For each test case, you will read an integer N ($1 \le N \le 8$) followed by N integers H_i ($1 \le H_i \le 100000$) where N is the number of pipes and H_i is the cross sectional height of i^{th} pipe.

Output

For each test case, print the case number and the desired volume of the box as shown in the sample output section. Any floating point errors less than 10^{-6} will be ignored by the judge.

Sample Input	Output for Sample Input
3 3 5 5 5 1 25 610 2 2 9 4 6	Case 1: 600 Case 2: 5000 Case 3: 2331.370849898476

Note: It is up to you how many digits you want to print after decimal point. Just make sure that the amour t of error does not exceed the specified limit.

G Gain Battle Power

The battle of Hogwarts is going to start very soon. Hermione has received some very important information about the death eaters. They have invented a new way to increase their **power** using their wands. Each death eater can carry two wands, one in the left hand and other in the right hand. They will stand in a line and create the front of their army. Hermione knows the order of the death eaters in the line and the value of their ***strength***. For death eaters, the value of ***strength*** and **power** may be different.

The value of **power** of each death eater is initially 1. They can use both of their wands to increase **power**. One can use his/her left hand's wand to connect to another death eater's left hand whose ***strength*** is strictly less than connector's ***strength*** and also in the left side of the connector. Same is true for right hand i.e. one can use his/her right hand's wand to connect to another death eater's right hand whose ***strength*** is strictly less than connector's ***strength*** and also in the right side of the connector. In this way, s/he can create a sequence where the strength increases from the leftmost person, becomes highest at his/her position and decreases on the right side. The **power** is equal to the length of this sequence and become fixed for the rest of the war. Each death eater will maximize his/her **power**.

After they fix their **power**, the war starts. Hermione and other members of the Order of Phoenix want to fight them individually (i.e. duel), but to do that they need to perform a special spell which splits the line or any segment of the line into two parts. The cost of performing this spell is equal to the sum of the **power** of the death eaters in that segment. Say, there are 3 death eaters and their **power** are 2, 1, 2 (**Sample Case 2**). Now if the splitting spell is performed between 1st and 2nd death eater, the 1st one becomes alone and 2nd and 3rd one are still together. So, in this case, if the first splitting spell is performed between 1st and 2nd, the cost is 2 + 1 + 2 = 5. Then the 2nd spell has to be performed between 2nd and 3rd death eater, which will cost 1 + 2 = 3. So the total cost is 8.

Hermione needs your help to **minimize** the total cost of splitting spells to make each death eater alone.

Input

First line of the input contains a positive integer, **T (T≤300)** which denotes the number of test cases. For each case, the first line contains the number of death eaters, **n (1≤n≤1000)**. The second line contains **n** positive integers denoting the **strength** of death eaters in the line (**Left to Right** i.e. **i-1** is on the left side of **i** and **i+1** is on the right side of **i** and (**2≤i≤n-1**). All integers are less than **1,000,000**.

Output

For each of the cases output "**Case <x>: <y>**" in a separate line, where **x** is case number, **y** is minimum total cost to break the union of death eaters.

Sample Input	Output for Sample Input
2 3 4 5 2 3 4 2 5	Case 1: 10 Case 2: 8

Explanation for Sample Case

In the first case, 1st death eater can make a sequence like 4 2 (no smaller strength in the left side). 2nd death eater can make a sequence 4 5 2, increasing in the left side **4 5** and decreasing in the right side **5 2**. 3rd death eater can make a sequence **2** (no smaller strength in the left side and also no smaller strength in the right side). So powers of them are 2, 3 and 1. To break the union of death eater one can perform splitting spell between 2nd and 3rd death eater which will cost 2 + 3 + 1 = 6 and then perform splitting spell between 1st and 2nd death eater which will cost 2 + 3 = 5. So the total cost is 6 + 5 = 11. But if one perform splitting spell between 1st and 2nd death eater it will cost 2 + 3 + 1 = 6 and then perform splitting spell between 2nd and 3rd death eater it will cost 3 + 1 = 4. So the total cost is 6 + 4 = 10 and it is minimum cost to break the union of death eaters.

H Hasmot Ali Professor

Professor Hasmot Ali loves to play string related problem. He assigns an easy lab task to his students. But they think it's a hard problem. I know you are very smart. You can help his students to solve this problem.

Given a string S, containing only lowercase English letters. There will be Q queries. Each line of query will contain two space separated strings, X and Y. For every query, your task is to calculate, how many distinct substrings of S which start with X and end with Y.

[Substring definition: A substring is any contiguous portion of a string. A substring may be empty, or the entire string]

For Example:

Given a string S = "abab". There are total 8 distinct substrings. The list is below:

[0] = "a"
[1] = "ab"
[2] = "aba"
[3] = "abab"
[4] = "b"
[5] = "ba"
[6] = "bab"
[7] =""
There are 3 queries:

1st Query: X = "a" and Y = "a".

There are 2 distinct substring of S, satisfy the condition ([0] = "a" and [2] ="aba").

2nd Query: X = "a" and Y ="b".

There are 2 distinct substring of S, satisfy the condition. ([1] = "ab" and [3] = "abab").

3rd Query: X = "ba" and Y ="ab".

There is only one distinct substring satisfy the condition. ([6] = "bab").

Input

Input start with an integer **T** ($\leq$ 3), denoting the number of test cases.

Each case starts with a line containing string **S (1 $\leq$ length(S) $\leq$ 1000)**. The next line contains an integer **Q** ($1 \le Q \le 50000$). Each of the next **Q** line contains two strings **X** ($1 \le$ length(X) ≤ 10) and **Y**($1 \le$ length(Y) ≤ 10).

Output

For each query you have to print the number of distinct substring of S, which are start with **X** and end with **Y**.

Sample Input	Output for Sample Input
1 abab 3 a a a b ba ab	Case 1: 2 2 1

I Identity Redemption

Once upon a time, Byteland was full of brilliant and genius computer programmers and algorithm experts. Programming contest was a great field for the programmers to test and show their talent. Now-a-days programming contest have been dried out in Byteland. Everyone is busy with different types of software development, mobile application development and gaming contest. Different sponsor provider and ministry of ICT of Byteland is funding different project to develop applications but not inspiring programming contests.

On the other hand, honorable minister of road transportation ministry is having some trouble with highway repairment in Byteland, as all the highways has been seriously damaged again on last rainy season. People of different cities are demonstrating protest against him to repair all the highways of the country. But recently ministry of road transportation is facing money problems. So they have to think twice before repair a highway whether they will be able to finish the repairment with existing money. The minister of road transportation has come up with a plan. He will select a set of highways to repair such that no city will be connected with more than one repaired highway. He will select maximum possible highways to repair, but the money is the concern here and every highway has its own cost to be repaired. So he will select maximum possible highways to be repaired with total cost as minimum as possible.

The minister of road transportation ministry thought this problem can be solved using computer software. He passed the problem to the champions of several software development and app development contests through the minister of ICT ministry. They are trying to solve the problem from then but still could not succeed. It is your chance to solve the problem and show the minister of ICT ministry how much programming contest is needed in Byteland to regain their pride in Information Technology.

The country of Byteland consists of **N** cities and **M** highways, where each highway connects two different cities and has a cost to repair. Given the description of Byteland's road transportation, find the minimum possible cost to repair maximum number of highways where each city is incident with at most one repaired highway. Cost of repairing is the sum of cost of all the repaired highways.

Input

Input starts with an integer, **T (T ≤ 100)** denoting the number of test cases. Each case starts with two integers, **N (1 ≤ N ≤ 50)** and **M (0 ≤ M ≤ N*(N-1)/2)**. Each of the next **M** lines contains three integers **u, v** and **c (1 ≤ u, v ≤ N, u ≠ v, 1 ≤ c ≤ 10^9)** meaning that there is a highway between city **u** and **v** and **c** is the cost to repair the highway. No highway will be mentioned twice in a test case.

Output

For each case, print the test case number, starting from **1**, and the minimum cost to repair maximum possible highways with the constraint given above.

Sample Input	Output for Sample Input
2 4 3 1 2 5 2 3 6 3 4 7 3 3 1 2 100 1 3 200 2 3 300	Case 1: 12 Case 2: 100

J

Judge in Queue

The great judge Upo again lost his passport. So he went to the passport office to get a new one. As a lot of people come in passport office, there are always several long queues. People wait in the queue until they get the job done. While waiting in the queue for some time, Upo came up with the following problem! How about writing a program to determine the best way to arrange **N** peoples into **M** queues, so that the **waiting time of a person who waits longest is minimized**?

You are given an array **W** with **N** elements, denoting the time **N** people already waited standing in the queue. An array **T** with **M** integers, where **T[i]** is the time needed to serve a single person in i^{th} queue. As Upo is standing in the queue right now. He wants you to right the program for him, to calculate the minimum waiting time for a person who waits the longest.

Input

Input starts with **T**, the number of test cases to follow. First line of each test case is two integers **N** and **M**. In the next line will contain **N** space separated integer representing the time **N** people already waited. Next line contains **M** space separated integer where i^{th} integer is the serving time of a single person in i^{th} queue.

$1 \le T \le 10, 1 \le N \le 10^5, 1 \le M \le 10^4, 1 \le W[i] \le 10^9, 1 \le T[i] \le 10^9$

Output

For each test case output the test case number starts from 1 and a number denoting the minimum waiting time for a person who waits the longest.

Sample Input	Output for Sample Input
1 4 2 3 1 2 1 2 3	Case 1: ~~2~~ 4

Problemsetter's Credits

Abdullah Al Raqibul Islam	Therap (BD) Ltd.
Abu Obaida Opu	SDSL
Anindya Das	Iowa State University
Anna Fariha	University of Dhaka
Arifuzzaman Arif	Google Inc.
F. A. Rezaur Rahman Chowdhury	TigerIT Bangladesh
Forhad Ahmed	TigerIT Bangladesh
Md. Shiplu Hawlader	University of Dhaka
Mohammad Hafiz Uddin	Eyeball Networks
Shafaet Ashraf	National University of Singapore
Zobayer Hasan	TigerIT Bangladesh

You can print your documents, assignments, lab reports, projects reports etc from Here at lowest cost.

Printing Cost (Black & White):
Normal: 1.80/- (Single), 2.20/- (Both Side)
Good:-- 2.0/- (Single), 2.40/- (Both Side)

Printing Cost (Color):
Normal: 2.50/- (Single), 3.00/- (Both Side)
Good:-- 2.70/- (Single), 3.20/- (Both Side)