A AKU NEGARAKU

Input	Standard Input
Output	Standard Output
Time Limit	3 seconds

Problem Description

1st Academy is an international leadership training academy based in Kuala Lumpur. Every year, the company trains thousands of people to be supplied to companies around the world. To be fair amongst all the trainees, the company will do the selection process using numbering system. The trainees will choose a number from 1 to *N*, and one number is not limited to only one trainee. The *N* represents the total number of companies that request trainees from the academy. A number, *M*, would be picked at random, and the selection starts with a trainee whose number is 1, and then in every *M*'th people after that, wrapping around to 1 after *N*, and ignoring trainees already selected. For example, if $N = 15$ and $M = 6$, trainees would be selected in the order: 6, 12, 3, 10, 2, 11, 5, 15, 13, 9, 14, 4, 1, 8, 7. All the selected trainees except the last one (which is number 7) will be supplied to companies outside of Malaysia.

However, Leong preferred to stay and work in Malaysia. To him, there is no better place other than Malaysia. He does not mind being located anywhere as long it is Malaysia. As a friend of him, you could help him to choose a number that will save him from being located outside.

Input

Input will consist of a series of lines, each line containing the number of companies (*N*) with $1 \leq N \leq 1500$, and a skipping value (*M*) with $1 \leq M \leq 50$. The values will be terminated by a line consisting of double zeros (0 0) as shown in sample input output.

Output

Output will consist of a series of lines, one for each line of the input. Each line will consist of the number *M* according to the above scheme.

Sample Input Output

Sample Input	Sample Output
15 6 550 23 0 0	7 470

B	CHEAP DELIVERIES	
	Input	Standard Input
	Output	Standard Output
	Time Limit per test case Memory limit per test case	2 seconds 64 megabytes

Problem Description

Abu runs a delivery service where he deliver items from one city to another. As with any business, Abu wants to decrease his cost as much as possible. The further he travel, the more fuel he will use.

In any particular day, Abu have k items to deliver. Each item needs to be delivered from a start city to a destination city. Each city is represented by an integer. Because of his business policies, he can only deliver one item at a time. However, he can deliver the items in any order that he wants, as long as he deliver all of them. So, everyday he starts at an item's start city and deliver the item to its destination city. Then, he goes to the next items's start city and deliver the item to the its destination city. And, he does this until he does not have any item left to deliver.

From experimentation, Abu notices that the distance he travels change if he change the order of his delivery. He thought, he can save a lot of money if he knows the best delivery order. He knows that you are very good at solving this kind of problem. So he wants you to solve it for him. Given a list of cities, a list of roads between the cities (and the road's length), and a description of deliveries he must do, determine what is the minimum total travel distance, given that he execute his delivery in the most efficient order.

Every road is bidirectional and there can be more than one road between two cities. Abu can use any road as many time as he wants.

Input

The first line consists of two integer n, m, k ($2 \le n, m \le 10^4$), ($1 \le k \le 18$) which is the number of cities, the number of roads and the number of items respectively.

The next m line each consist of 3 integers, u_i, v_i, l_i ($1 \le u_i, v_i \le 10^4$), ($1 \le l_i \le 10^6$), which denotes that a road exists from city u_i to v_i with length l_i.

The next k line each consist of 2 integers, f_i, d_i ($1 \le f_i, d_i \le 10^4$) which denotes that the ith item is from city f_i and its destination is city d_i.

Output

A single integer, which is the minimum total travel distance given that Abu deliver all items optimally, or -1 if its is impossible for him to deliver all items.

Sample Input Output

Sample Input	Sample Output
5 5 3 1 2 1 2 3 2 3 4 3 4 5 4 5 2 4 2 3 1 2 5 3	12

Sample Input	Sample Output
5 5 4 1 2 10 5 3 10 2 4 1 4 1 2 3 5 4 1 2 3 5 4 1 2 4	-1

Note

In the first example, Abu can start from city 5, and deliver the third item, so he move to city 3, travelling a total distance of 6 by going through city 2, then he goes to city 1, travelling a distance of 3 through city 2, and then he deliver the first item to city 2 and then deliver the second item to city 3. The total travel distance is 12.

In the second example, city 1, 2 and 4 is disconnected from city 3 and 5, therefore it is impossible for Abu to deliver all item.

C	ELI'S CURIOUS MIND	
	Input	Standard Input
	Output	Standard Output
	Time Limit	3 second

Problem Description

Eli is a teenager who loves to study chemistry. She recently joined a chemistry research lab. Dr. Phil wants her to just play around with some chemicals and observe their reaction. Therefore, he gave her a one-row tray of test tubes with the different chemical inside of them and told her:

"Mix these chemical together anyhow that you like, but the you have to follow two rules:

1. Never make a mixture that has two chemicals that their tubes are next to each other.
2. Keep adding more chemical to the mixture until it is not violating the new rule. "

For example, in the image you can see she was given 5 tubes and she is able to make 4 different mixture without violating the rule: {1,3,5}, {2,4}, {2,5}, {1,4}.

But she cannot mix 1 and 3 only because she still can add 5 without violating the rules.

She is curious to know how many different mixtures she can make without violating the rule with any given number of tubes. That's why she asks you write a code to calculate it for her.

Input

The input will consist of a sequence of numbers N, $1 \leq N \leq 76$. Each number will be on a separate line. The input will be terminated by **0**.

Output

Output the number of different mixture she can make without violating the rule mentioned above on a single line as show in the sample. The number of all subsets will be less than 2^{31}.

Sample Input Output

Sample Input	Sample Output
1 2 3 4 5 30 0	Case #1: 0 Case #2: 0 Case #3: 1 Case #4: 3 Case #5: 4 Case #6: 4410

D	EXPLORACE	
	Input	Standard input
	Output	Standard output
	Time Limit	3 seconds

Problem Description

CodingSchool is conducting an explorace to welcome new students. It is compulsory for each team to visit all check points (not necessarily following the sequence). At each check point, the team will have to complete a specific activity. Each team can plan a strategy on the sequence of check points to visit. The distance of each path is no more than 500 meters.

Because they don't want the new student to wander away and get lost, CodingSchool wants to put their committee on the paths and only allow the student to use path that have a committee. But CodingSchool only have a limited number of committee, so they don't want to use all path. Shorter path is preferred because it use less committee. While at the same time, they must make sure that there exists a way to travel between every two checkpoints. Help CodingSchool by determining the minimum total distance of path that they must cover.

Input

First line of input is integer T ($1 \leq T \leq 10$) that represents the number of test cases. Each test case starts with a line with two integers $\boldsymbol{N}$ ($1 \leq N \leq 20$) and $\boldsymbol{M}$ ($1 \leq M \leq N*(N-1)$), that represents the number of check points and the number of paths to consider respectively. In the following $\boldsymbol{M}$ lines, there are 3 integers a, b ($0 \leq a, b <$ N) and d ($1 \leq d \leq 500$) that represent the start check points (a), the end check points (b) and the distance of the path (d) that connects check points a and b.

Output

For each test case, output the minimum distance as shown in the sample output.

Sample Input Output

Sample Input	Sample Output
3 5 7 1 2 75 2 3 32 3 4 62 1 4 50 3 5 100 4 5 45 1 5 78 6 8 1 2 70 1 4 82 2 3 57 2 5 105 3 4 160 3 6 55 4 5 97 4 6 75 4 5 1 2 10	Case #1: 189 meters Case #2: 354 meters Case #3: 70 meters

1 3 30 1 4 40 2 3 20 3 4 50	

E MATRIX MULTIPLICATION CALCULATOR

Input	Standard Input
Output	Standard Output
Time Limit	2 seconds

Problem Description

Matrix multiplication is a basic tool of linear algebra, and has numerous applications in many areas of mathematics, as well as in applied mathematics, computer graphics, physics, and engineering.

We can only multiply two matrices if their dimensions are compatible, which means the number of columns in the first matrix is the same as the number of rows in the second matrix.

If A = $[a_{ij}]$ is an m×n matrix and B = $[b_{ij}]$ is an n×q matrix, the product AB is an m×q matrix. The product AB is defined to be the m×q matrix C = $[c_{ij}]$ such that

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Your task is to design a matrix multiplication calculator to multiply two matrices and display the output. If the matrices cannot be multiplied, display "`undefined`"

Input

The input consists of a few test cases. For each test case, the first line of input is 4 positive integers, M, N, P and Q ($1 \leq M, N, P, Q \leq 20$). M and N represent the dimension of matrix A, while P and Q represent the dimension of matrix B. The following M lines consist of the data for matrix A followed by P lines that contains the data for matrix B as shown in the sample input. The test data ends when M, N, P and Q are 0.

Output

For each test case, the output contains a line in the format "`Case #x:`", where x is the case number (starting from 1). The following line(s) is the output of the matrix multiplication.

Sample Input Output

Sample Input	Sample Output
2 3 3 2 1 2 3 3 2 1 4 5 6 7 8 9 2 3 2 3 1 2 3 3 2 1 4 5 6 7 8 9 0 0 0 0	Case #1: | 40 46 | | 32 38 | Case #2: undefined

F SUM OF SUB RECTANGLE AREAS

Input	Standard Input
Output	Standard Output
Time Limit	2 second

The following code snippet calculates the sum of the areas of all the sub rectangular grid in a rectangular grid from (0,0) to (N,N). Find an efficient way to compute that.

```
sum = 0
for r1 = 0 to N-1
  for c1 = 0 to N-1
    for r2 = r1+1 to N
      for c2 = r2+1 to N
        sum = sum + (r2-r1)*(c2-c1)
print(sum)
```

Input

Input starts with T the number of test cases. Each test case consists of a single integer N.

Output

For each test output the sum (as computed above). Please note that even though W and H will fit into 64 bit integer, the sum may not be.

Sample Input Output

Sample Input	Sample Output
5 1 2 3 4 1000	1 16 100 400 27944805889000000

G	WAK SANI SATAY	
	Input	Standard Input
	Output	Standard Output
	Time Limit	2 second

Problem Description

Wak Sani Satay is a humble stall located in Kajang and has been around since 1969. Many like Wak Sani's satay because the meat is juicy and tender, served with creamy and sweet *kuah kacang*, alongside with *nasi impit*, cucumber and onion slices.

Wak Sani usually calculates his net profit at the end of the week. The net profit is calculated by subtracting the cost from the gross profit. He can get 85 sticks of satay from 1 kg meat. The price for 3 types of satay are shown in Table 1. The price for *nasi impit* is RM0.80 each while cucumber and onion slices are free of charge.

The cost for making satay for each meat type in shown in Table 2. The cost of spices to marinate satay for every kilogram of meat is RM8.00 and the cost for each *nasi impit* is RM0.20 each.

Table 1

Satay	**Price per stick**
Chicken	RM0.80
Beef	RM1.00
Lamb	RM1.20

Table 2

Meat	**Price per kg**
Chicken	RM7.50
Beef	RM24.00
Lamb	RM32.00

Write a program to find the weekly net profit.

Input

The input consists of a few sets of test cases. The first line for each data case is an integer N ($1 \leq N \leq 7$), which represents the number of days the stall is opened to customers for a week. It is followed by N lines of data, each line represents the sales (in sticks) of chicken satay, beef satay, lamb satay and *nasi impit* per day. Input is terminated by a test case where N is 0.

Output

For each test case, output a line in the format "`Case #x: RM`" where `x` is the case number (starting from 1), follow by the calculated net profit in Malaysian currency format as shown in the sample output.

Sample Input Output

Sample Input	Sample Output
1 30 40 34 5 2 0 0 0 0 1 1 1 1 3 1000 1000 1000 10 5000 3000 4000 12 100 300 10 6 0	Case #1: RM71.27 Case #2: RM2.57 Case #3: RM10119.98

H

STROOP EFFECT

Input	Standard Input
Output	Standard Output
Time Limit	2 seconds

Problem Description

The Stroop effect is a phenomenon that occurs when you must say the color of a word but not the name of the word. When the name of a color (e.g., "blue", "green", or "red") is printed in a color that is not denoted by the name (e.g., the word "red" printed in blue ink instead of red ink), naming the color of the word takes longer and is more prone to errors than when the color of the ink matches the name of the color. The effect is named after John Ridley Stroop, who first published the effect in English in 1935. The original paper has been one of the most cited papers in the history of experimental psychology, leading to more than 700 replications. The effect has been used to create a psychological test (Stroop test) that is widely used in clinical practice and investigation.

Stimuli in Stroop paradigms can be divided into 2 groups: congruent and incongruent. Congruent stimuli are those in which the ink color and the word refer to the same color (for example the word "red" written in red ink). Incongruent stimuli are those in which ink color and word differ (for example the word "blue" written in red ink). There are four primary color in Stroop test namely; blue, yellow, red and green. In Stroop test, a number of stimuli from both group will be randomly displayed, one by one. Each time it is displayed, the patient need to respond on the correct color, not the word. In addition, the number of stimuli group in Stroop test are the same for both congruent and incongruent. The number of each color in congruent group and incongruent group must be the same. In addition, the combination of word-color in incongruent group must be the same. must be the same, Also, only one consecutive same ink color or word color are allowed.

For example, stimuli frequency is as follows:

Word-Color	Color=blue	Color=yellow	Color=green	Color=red
Word=blue	3	1	1	1
Word =yellow	1	3	1	1
Word =green	1	1	3	1
Word =red	1	1	1	3

Overall number of stimuli is 24. Number of congruent stimuli equals to incongruent stimuli that is 12. Incongruent stimuli for each color of each word is the same that is 1.

Stimuli sequence is invalid if exist more than one consecutive same color or same word, eg: … blue-blue, green-blue, red-blue ...

Given a sequence of two digits, your task is to identify whether the sequence fulfills Stroop test stimulus. The digit 1 – 4 represents four primary color, the first digit represent the word color, and the second digit represent the ink color.

Input

The first line of input is an integer N ($1 \leq N \leq 50$) that represents the number of test case, followed by N lines where each line of input contains a sequence of two digit integers 1, 2, 3 or 4. The line end with number 00. The sequence length is not more than 500.

Output

For each test case, the output contains a line in the format *Case #x: result*, where x is the case number (starting from 1) and *result* is either *Stroop* or *Not Stroop*.

Sample Input Output

Sample Input
2 11 42 14 34 11 13 33 41 23 22 11 21 22 31 44 32 33 12 44 22 44 43 24 33 00 11 34 00
Sample Output
Case #1: Stroop Case #2: Not Stroop

I SUPER BALL

Input	Standard Input
Output	Standard Output
Time Limit per test case	2 seconds
Memory limit per test case	256 megabytes

In the distance future, there exist a company that makes ball. Technically they make spherical object regardless of use, but let's just call it a ball. Their ball consist of many layers wrapped around each other. Maybe the first layer is made of metal, second made of rubber, third made of carbon nanotubes, the forth layer maybe a single-atom-thick graphene. Think of it like an onion. These layers are so advance that each type of layer have their own factory. Logistic becomes a headache because during the production of the ball, the ball needs to be transferred from one factory to another factory in order of their type of layer. For example, a ball's core (the first layer) is made in the metal factory, and then it is transferred to the rubber factory and then it is transferred to the carbon nanotubes factory and then to the graphene factory.

Fortunately, some factory can make more than one type of layer, so in that case, the ball may not need to be transferred to another factory. Plus, there can be more than one factory that can make a particular type of layer, so if its cheaper to use that factory, the company can. Unfortunately, because this is the future, absolutely everything must be recycle. And because the ball are so advance, each layer of the ball can only be recycled by the factory that can make that layer.

Because this is a company that wants to make money, they need to know how much a ball will cost and they want to minimize it. That includes the cost of making a layer, the cost of recycling a layer and the cost of transferring the ball from one factory to another. Thankfully, delivery to and from the client is handled by another company and they pick up the ball from any factory to the client and send it back for recycling to any factory for free. As you can imagine, when the manager receive an order of a ball made of 101 exotic layer, some of his brain cell commit suicide. So he turns to you to make his life easier. Given a description of the factories, and a description of a type of ball calculate the minimum cost to produce the ball.

Input

The first line consist of two number F ($1 \le F \le 500$) which is the number of factory and L ($1 \le L \le 500$) which is the number of type of layer. Each factory is represented by an integer from 1 to F and each layer type is represented by an integer from 1 to L.

The next $3F$ line represent the factory information, each factory is described with three line.

The first line consist of F integers, C_i($0 \le C_i \le 10^3$) where C_i is the cost of transferring a ball from the current factory to the i'th factory.

The second line consist of L integer, $D_i(-1 \le D_i \le 10^3)$ where D_i is the cost of producing the layer type i.

The third line consist of L integer, $R_i(-1 \le R_i \le 10^3)$ where R_i is the cost of recycling the layer type i.

If D_i or R_i is -1, that means the factory cannot produce or recycle the layer type.

The next line represent the ball layer configuration.

The line start with an integer $N(1 \le N \le 500)$ which is the number of layer in the ball. The next N integer, $G_i(1 \le G_i \le L)$ represent the type of layer of the ball.

Output

Output a single integer R which is the minimum cost of producing the ball.

Sample Input Output

Sample Input	Sample Output
3 3 0 10 15 99 -1 -1 10 -1 -1 10 0 5 -1 10 10 -1 5 5 15 5 0 -1 1 -1 -1 20 -1 2 3 2	26

Sample Input	Sample Output
3 3 0 10 15 99 -1 -1 10 -1 -1 10 0 5 -1 10 10 -1 5 5 15 5 0 -1 1 -1 -1 20 -1 5 1 2 3 2 1	303

Note

In the first example, the ball configuration is 3, 2. The cheapest way to make and recycle the ball is from factory 3, costing 1 for first layer, then, factory 2, with transfer cost of 5 and produce costing of 10 for the second layer. The for recycling, the second layer is recycled starting at factory 2, costing 5 for second layer and 5 for the first layer, again at the same factory (so no transfer cost).h

J VIRUS OUTBREAK

Input	Standard Input
Output	Standard Output
Time Limit	2 seconds

Problem Description

The State Veterinary Services Department recently reported an outbreak of a newly found cow disease. All cows found to have affected by the disease have since euthanized because of the risk to the industry. Number of affected cows increased to 21, 34 and reached 55 after eight, nine and ten hours respectively.

You have been assigned by the authority to study the pattern of the outbreak and develop a program to predict the next number of affected cows so that the authorities could prepare and act accordingly.

Input

Input will consist of a series of positive integer numbers no greater than 490 each on a separate line represent the hours. The input process will be terminated by a line containing -1.

Output

For each input value, the output contains a line in the format: **Hour X: Y cow(s) affected**, where X is the hour, and Y is the total affected cows that need to be euthanized based on the hour given by X.

Sample Input Output

Sample Input	Sample Output
1 4 6 11 -1	Hour 1: 1 cow(s) affected Hour 4: 3 cow(s) affected Hour 6: 8 cow(s) affected Hour 11: 89 cow(s) affected