

Problem A. Everyone Loves Playing Games

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 256 mebibytes

One day, Acesrc and Roundgod are playing an interesting game called Important Choice Pairs of Cakes (ICPC).

In this game, there is a variable X . Initially, X is equal to 0. N pairs of numbers (x_i, y_i) are given to Acesrc while M pairs (x'_i, y'_i) are given to Roundgod.

Firstly, for every pair (x_i, y_i) , Acesrc will choose either x_i or y_i . Suppose he choose k , X will be changed to $(X \oplus k)$. (\oplus denotes bitwise exclusive or)

After Acesrc's N operations, Roundgod will do the same with his M pairs.

They know about each other's pairs from the beginning. Acesrc wishes the final value of X to be as great as possible while Roundgod wishes it to be as small as possible.

Acesrc and Roundgod are very clever boys and they will choose the best strategy. Can you predict the final value of X ?

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 20$), indicating the number of test cases. For each test case:

The first line contains two integers N and M ($1 \leq N, M \leq 10000$).

Then N lines follow. In each line, there are two integers x_i, y_i ($1 \leq x_i, y_i \leq 10^{18}$), representing Acesrc's pairs.

Then M lines follow. In each line, there are two integers x'_i, y'_i ($1 \leq x'_i, y'_i \leq 10^{18}$), representing Roundgod's pairs.

Output

For each test case, you should output a single integer in a line as your answer.

Example

standard input	standard output
2	2
1 1	2
6 3	
4 1	
2 2	
1 3	
4 6	
5 4	
2 2	

Note

In the first sample, if Acesrc chooses 6, Roundgod will choose 4 and the result will be 2.

If Acesrc chooses 3, Roundgod will choose 1 and the result will also be 2.

Therefore the answer is 2.

Problem B. Gifted Composer

Input file: *standard input*
 Output file: *standard output*
 Time limit: 4 seconds
 Memory limit: 256 mebibytes

Acesrc is a gifted composer. He likes writing tuneful and melodic songs. Every song he writes can be viewed as a sequence of musical notes, and each musical note represents the pitch and the duration of the sound. In this problem, we consider only the following seven primary pitches

do re mi fa sol la si

and the duration of each note is one unit time. Hence, there are only seven types of notes, and we may use the pitch name to represent a note.

Acesrc composes a song in the following way. Initially, the sequence of notes is empty. Every day, he inserts a new note at the beginning or at the end of the sequence, until the song is done.

Acesrc particularly likes songs with repetitions. For a song with n musical notes, we say the song has a repetition of length k ($1 \leq k \leq n$), if the song can be partitioned into one or more identical sections with k notes, optionally followed by an incomplete section, which is an initial part of a complete section. For example, do re do re do can be partitioned into do re | do re | do, so it has a repetition of length 2; similarly, do re mi do re mi has a repetition of length 3, and do re do re mi has a repetition of length 5.

Acesrc wants to know, after he adds a note each day, the number of different lengths of repetitions the song has. Can you help him?

Input

The first line of input consists of a single line n ($1 \leq n \leq 10^6$), the number of days Acesrc uses to compose the song. The i th of the remaining n lines contains a character a ($a \in \{p, a\}$) (where p denotes prepend, i.e., inserting at the beginning, and a denotes append, i.e., inserting at the end) and a string s ($s \in \{do, re, mi, fa, sol, la, si\}$), representing the action Acesrc takes in the i th day.

Output

Output n lines. The i th line should be a single integer, denoting the answer for the i th day.

Examples

standard input	standard output
5 a do p re a re a do p do	1 1 2 2 3
5 a re a do a re p do a mi	1 1 2 2 1

Problem C. An Unsure Catch

Input file: *standard input*
Output file: *standard output*
Time limit: 8 seconds
Memory limit: 256 mebibytes

Some prisoners have managed to escape from the prison! Your job is to get them back.

At time 0, there are n prisoners out in the city, each at a different position. We label these positions from 1 to n .

Your colleagues have made a plan to block as many road as possible so that there will be only one way out for each position. Once in an hour, the prisoners at position i will leave and arrive at position a_i ($1 \leq a_i \leq n$) immediately.

You can arrange an assault at any integer time at any position. Every prisoner at the same location at that time will be arrested. However, you can only attack once, after that all the prisoners on the run will leave the city.

After consideration, you decide to contact your colleagues to change their plan. Specifically, you can change some a_i before time 0 ($1 \leq a_i \leq n$) must still hold after changing).

Now you want to know, what's the minimum number of a_i to change (let's denote the answer as K) in order to catch all the prisoners. You also want to know, if you can change a_i for at most j ($j \in \{0, 1, 2, \dots, K\}$) times, how many prisoners at most can you arrest. Write a program to solve the problem.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 10^4$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 10^5$), indicating the number of prisoners and positions.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$), their meanings are described above.

For only about 5 cases will n larger than 50.

Output

For each test case output two lines.

For the first line, print an integer K , indicating the least number of a_i to change so that you can arrest all the prisoners.

For the second line, print $K + 1$ integers separated by one space, indicating the number of prisoners you can arrest at most if you can change a_i for at most $0, 1, 2, \dots, K$ times.

Example

standard input	standard output
2	0
1	1
1	3
10	3 6 9 10
2 3 1 4 5 7 6 6 6 7	

Problem D. String Theory

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

Acesrc is a famous string theorist at Nanjing University second to none. He insists that we should always say an important thing k times. He also believes that every string that can be obtained by concatenating k copies of some nonempty string is splendid. So, he always teaches newcomers, “String theory problems are important! String theory problems are important! ... String theory problems are important!”

Today, he wants to examine whether the newcomers remember his instruction. He presents a string consisting of lower case letters and asks them the number of splendid substrings of the presented string. No one can solve this problem, and they will be scolded for hours. Can you help them solve this problem?

Note that equal splendid substrings occurred in different positions should be counted separately.

Input

The first line of input consists of a single integer T ($1 \leq T \leq 10$), denoting the number of test cases. Each test case starts with a single line of an integer k ($1 \leq k \leq 20$). The second line of a test case is a string S consisting of lower case letters only, the length of which is between 1 and 3×10^5 inclusive. The sum of the lengths of strings in all test cases never exceeds 10^6 .

Output

For each test case, print the answer as a single integer in a single line.

Example

standard input	standard output
3	2
2	6
aabb	0
2	
abababab	
3	
abc	

Problem E. Road Construction

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 256 mebibytes

There are $n + m$ towns in Kingdom of Coffee Chicken, which can be seen as $n + m$ integers coordinates (x_i, y_i) on the 2-dimensional plane. n of them belong to Acesrc while the other m towns belong to Roundgod.

Now both Acesrc and Roundgod want to build straight roads among their towns and they all want their towns are connected, which means there is a path between any two of towns. It is obvious that we need only $n + m - 2$ roads to make it possible. Moreover, Acesrc and Roundgod hope that among these $n + m - 2$ roads, there is no intersection other than the position of towns.

Now we hope you to provide us a construction plan.

Input

The first line contains two integers $n, m (n > 1, m > 1, n + m \leq 3000)$.

The following n lines describe Acesrc's towns and each line contains two integers $x, y (0 \leq x, y \leq 10^9)$ representing coordinates. Their number is $1 - n$ respectively.

The following n lines describe Roundgod's towns and each line contains two integers $x, y (0 \leq x, y \leq 10^9)$ representing coordinates. Their number is $1 - m$ respectively.

There is no repeated coordinates among those $n + m$ towns. We also guarantee that no three towns are on the same straight line among them.

Output

Please output $n + m - 2$ lines in total, the first $n - 1$ lines representing the construction plan of Acesrc's towns and the other $m - 1$ lines representing the construction plan of Roundgod's towns. For each line of a construction plan, please output two integers x, y , indicating a straight road connected town x and y .

If it is impossible to find any valid construction plan, output **Impossible** instead.

Example

standard input	standard output
2 3	2 1
0 0	1 3
1 1	3 2
1 0	
0 1	
2 3	

Problem F. Girlfriend

Input file: *standard input*
 Output file: *standard output*
 Time limit: 7 seconds
 Memory limit: 256 mebibytes

Acesrc and his girlfriend are famous sweet lovers at Nanjing University second to none. They often wander along the streets near Xinjiekou, and taste takoyaki.

Xinjiekou is the central business district of Nanjing, consisting of n crossroads and m streets connecting some pairs of these crossroads. By every street there is a takoyaki bar, and Acesrc has a preference value for each of these bars. The streets are bidirectional.

Acesrc and his girlfriend choose a pair of crossroads as the starting point and the ending point; moreover, they choose a route between these two crossroads, and they walk along the streets in the route. Every crossroad appears in the route at most once. Once they encounter a takoyaki bar they will buy a serve of takoyaki from this bar. However, they can keep at most two serves of takoyaki. If, after buying a serve of takoyaki they have more than two serves, they will discard one serve with minimum preference value.

After reaching the ending point they begin to enjoy the takoyaki. Acesrc always eats the serve with lower preference value; however, if they only bought one serve, poor Acesrc will have nothing to eat. The clever Acesrc wants to know the minimum possible preference value of the serve he eats, given the starting point and the ending point of their route. Can you tell him the value?

Input

The first line of input consists of a single integer T ($1 \leq T \leq 100$), the number of test cases.

For each test case, the first line consists of three integers n, m, q ($1 \leq n, q \leq 10^5, 1 \leq m \leq 2 \times 10^5$), denoting the number of crossroads, streets in Xinjiekou, and the number of queries, respectively. Each of the next m lines contains three integers x, y, w ($1 \leq x, y \leq n, x \neq y, 1 \leq w \leq 10^9$), denoting a street connecting the x th and the y th crossroads, with a takoyaki bar of preference value w near the street. It is possible that multiple streets connect the same pair of crossroads. Each of the last q lines are two integers u, v ($1 \leq u, v \leq n, u \neq v$), specifying a query that the starting point and ending point are the u th and the v th crossroads, respectively.

It is guaranteed that the sum of n and the sum of q over all test cases do not exceed 2.5×10^5 , and the sum of m does not exceed 5×10^5 .

Output

For each query of each test case, print the answer in one line. If Acesrc might have nothing to eat, print 0; if there is no path between the given starting and ending point, print -1.

Example

standard input	standard output
1	0
4 2 3	2
1 2 2	-1
2 3 3	
1 2	
1 3	
1 4	

Problem G. Blackjack

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 256 mebibytes

As the saying goes, “if you are hesitant, you will lose the game; if you are decisive, you will die in vain.” Calabash doesn’t want to lose the game, neither to die in vain. But, this is not important, because he doesn’t have enough money to buy the game.

Fortunately, Roundgod, the banker of a casino, gives him a chance to fulfill his dream. Roundgod promises to buy him the game if he could win in one round of Blackjack. The Blackjack involves a deck of playing cards, each has some points on it. It is carried out as follows: Roundgod gives a number a and tells Calabash the number of points on each of the cards. Then, Calabash begins to draw cards, according to the following procedure:

1. the deck of all cards is shuffled uniformly at random;
2. Calabash could repeatedly draw a card from the deck (he immediately knows the number of the points on the card after drawing the card), or declare to stop drawing at any moment;
3. if, after drawing any card, the total number of points on cards Calabash drawn exceeds a limit b , he loses immediately;
4. otherwise, after Calabash stops drawing, he wins if and only if the total number of points on cards he drew is greater than a .

Calabash wants to know the probability to win if he plays optimally so that he can obtain his favorite game. Please help him to calculate the probability.

Input

There are two lines in the input. The first line consists of three integers n, a, b ($1 \leq n \leq 500, 1 \leq a < b \leq 500$), denoting the number of cards in the deck, the number given by Roundgod, and the limit to the total number of points, respectively.

The second line contains n integers x_1, x_2, \dots, x_n ($1 \leq x_i \leq 500$), denoting the numbers of points of these cards. It is guaranteed that $\sum_{i=1}^n x_i > b$.

Output

Output the answer within an absolute error of no more than 10^{-6} .

Examples

standard input	standard output
5 2 4 1 1 1 5 5	0.100000000000
5 2 4 1 1 1 3 5	0.450000000000

Problem H. Yet Another Geometry Problem

Input file: *standard input*
 Output file: *standard output*
 Time limit: 4 seconds
 Memory limit: 256 mebibytes

There is a 2-dimensional plane described as $\{(x, y) | 0 \leq x \leq M, 0 \leq y \leq M\}$. We also have another N points $P(x_i, y_i)$. Different points may share the same coordinates.

We define a good space as a square(in the given plane) with no point strictly inside it. Endpoints of the square should be on integers coordinates.

In each query, given (u, v) , please calculate the largest area of a good space which (u, v) is strictly inside.

Notice that the border of a legal space has to be parallel to x-axis or y-axis and it should not cross the border of the plane.

Input

There are multiple test cases. The first line of the input contains an integer $T(T \leq 10)$, indicating the number of test cases. For each test case:

The first line contains two integers $M(2 \leq M \leq 10^9)$ and $N(0 \leq N \leq 5000)$.

In the following N lines, each line contains two integers $X_i, Y_i(0 \leq X_i, Y_i \leq M)$, which denotes the Euclidean coordinate of $P(x_i, y_i)$.

Then the next line contains one integer $Q(1 \leq Q \leq 5000)$, which denotes the number of queries.

In the following Q lines, each line contains two integers $u, v(0 \leq u, v \leq M)$.

Output

For each query, please output an integer as the answer in one line.

Specially, if there is no legal good space, please output 0 instead.

Example

standard input	standard output
1	4
5 5	9
1 4	4
2 1	
3 2	
4 1	
4 4	
3	
3 1	
2 3	
4 3	

Problem I. A Math Problem

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

There are n fans $F_i (i = 1, \dots, n)$ and m teams $T_j (j = 1, \dots, m)$.

- (i) For any fan F_i , F_i is a fan of at least one team but not a fan of all teams.
- (ii) For any two teams $T_i, T_j (1 \leq i, j \leq m)$, there exists exactly one team $T_k (1 \leq k \leq m)$ exactly having the fans both T_i and T_j have. Note that i, j, k can be the same.
- (iii) For any two teams $T_i, T_j (1 \leq i, j \leq m)$, there exists exactly one team $T_k (1 \leq k \leq m)$ exactly having the fans either T_i or T_j have. Note that i, j, k can be the same.

Please calculate that How many kinds of correspondences between the fans and the teams.

Input

There are multiple test cases. The first line of the input contains an integer $T (T \leq 100000)$, indicating the number of test cases. For each test case:

The first and only line contains two integers $n, m (1 \leq n \leq 10^{18}, 2 \leq m \leq 6)$.

Output

For each test case, output a integer representing the answer modulo $1000000007(10^9 + 7)$ in one line.

Example

standard input	standard output
9	2
2 2	12
2 3	36
3 3	216
3 4	1032
4 4	7200
4 5	46800
5 5	453600
5 6	3369600
6 6	

Problem J. Gaokao

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Roundgod is about to attend Gaokao (National Unified Examination for Admissions to General Universities and Colleges) and his dream school is Zhejiang University. He sees an interesting problem while he is studying Math, which is a problem related to Pascal's Triangle.

The definition of Pascal's Triangle is given below:

The first element and the last element of each row in Pascal's Triangle is 1, and the m_{th} element of the n_{th} row equals to the sum of the m_{th} and the $(m - 1)_{th}$ element of the $(n - 1)_{th}$ row. Here's an example of a 5 levels Pascal's Triangle .

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

In the task, Roundgod is required to calculate how many elements in the 126_{th} row of Pascal's Triangle are odd numbers.

After solving it, Roundgod thinks of a harder version of this problem. He gives you many requests about similar questions but the row number will be bigger. Please calculate that how many elements in the k_{th} row of Pascal's Triangle are odd numbers.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 500$), indicating the number of test cases. For each test case:

The first and only line contains an integer K ($K \leq 10^{18}$), indicating the required row number in Pascal's Triangle.

Output

For each test case, output the number of odd numbers in the k_{th} line.

Example

standard input	standard output
3	2
3	4
4	2
5	

Problem K. Data Structure

Input file: *standard input*
 Output file: *standard output*
 Time limit: 20 seconds
 Memory limit: 512 mebibytes

Andy is a famous data structure expert at Nanjing University second to none. One day he throws a plain dry data structure problem to his friends, but none of them can solve. How about you?

Given a tree rooted at node 1. Each node has a weight which is 0 initially. Define the distance between two nodes as the number of edges in the unique simple path between the two nodes. You need to perform these two types of operations:

- Type 1: given a, x, y, z , add z to the weights of all a 's descendants (including a itself) whose distances to a are y modulo x ;
- Type 2: given a , return the weight of node a .

Input

The first line of the input is a single integer T ($1 \leq T \leq 4$), the number of test cases.

Each test cases starts with two integers n, m ($1 \leq n, m \leq 300000$), denoting that there are n nodes (numbered 1 through n) in the tree and you need to perform m operations. The next line contains $n - 1$ integers, f_1, f_2, \dots, f_{n-1} ($1 \leq f_i \leq i$), specifying the edges of the trees; the i th integer denotes the parent of node $i + 1$. The next m lines describe the operations. Each line is either $1 \ a \ x \ y \ z$ ($1 \leq a \leq n, 1 \leq x \leq n, 0 \leq y < x, 0 \leq z \leq 500$), denoting an operation of type 1, or $2 \ a$ ($1 \leq a \leq n$), denoting an operation of type 2.

Output

For each operation of type 2 in each test case, print the answer in one line.

Example

standard input	standard output
1	5
5 5	0
1 1 2 1	
1 1 5 4 1	
1 1 4 1 5	
1 2 1 0 4	
2 3	
2 1	

Problem L. Landlord

Input file: *standard input*
 Output file: *standard output*
 Time limit: 1 second
 Memory limit: 256 mebibytes

Calabash is the servant of a landlord. The landlord owns a piece of land, which can be regarded as an infinite 2D plane.

One day the landlord set up two orthogonal rectangular-shaped fences on his land. He asked Calabash a simple problem: how many nonempty connected components is my land divided into by these two fences, both finite and infinite? Calabash couldn't answer this simple question. Please help him!

Recall that a connected component is a maximal set of points not occupied by the fences, and every two points in the set are reachable without crossing the fence.

Input

The first line of input consists of a single integer T ($1 \leq T \leq 10000$), the number of test cases.

Each test case contains two lines, specifying the two rectangles. Each line contains four integers x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 10^9, x_1 < x_2, y_1 < y_2$), where $(x_1, y_1), (x_2, y_2)$ are the Cartesian coordinates of two opposite vertices of the rectangular fence. The edges of the rectangles are parallel to the coordinate axes. The edges of the two rectangles may intersect, overlap, or even coincide.

Output

For each test case, print the answer as an integer in one line.

Example

standard input	standard output
3	3
0 0 1 1	4
2 2 3 4	2
1 0 3 2	
0 1 2 3	
0 0 1 1	
0 0 1 1	

Problem M. Travel Dream

Input file: *standard input*
 Output file: *standard output*
 Time limit: 3 seconds
 Memory limit: 256 mebibytes

Andy is an ordinary student at Nanjing University. One day, he found himself transmitted to a wonderland when he was dreaming. The wonderland consisted of several spots, with some bidirectional roads of various travel times connecting some pairs of them.

Attracted by the fascinating scenery, Andy wanted to visit some spots in the wonderland. He would like to select exactly k distinct spots and visit them one by one. After visiting the last spot, he must turn back to the first spot, because that was where his dream started. There must be a road between any two adjacent spots he selected, including the last one and the first one. Moreover, he wanted to maximize the total time spent in moving between the spots, so that he could better enjoy the beauties.

For example, in the first sample data, you may choose to start from spot 1, visit spots 3, 5, 2, and return to spot 1. The total time spent was 21 minutes, which turns out to be optimal.

However, finding such a traveling plan was too hard for Andy, so he wanted your help. Could you help him to find such an optimal plan?

Input

The first line of input consists of three integers n, m, k ($2 \leq n \leq 300, 1 \leq m \leq 300, 3 \leq k \leq 10$), denoting the number of spots and the number of roads in the wonderland, and the number of spots Andy would like to visit, respectively. The spots are numbered 1 through n .

The remaining m lines specify the roads between the spots. Each of these lines contains three integers u, v, t ($1 \leq u, v \leq n, u \neq v, 1 \leq t \leq 10^8$), representing a bidirectional road between spots u and v , and it took t minutes to travel along the road in either direction. It is guaranteed that there was at most one road between any pair of spots in the wonderland.

Output

Print the maximum total travel time in minutes as an integer in one line. If it is impossible to find any valid travel plan, output `impossible` instead.

Examples

standard input	standard output
5 7 4 1 2 2 1 3 3 2 3 4 4 3 1 5 3 7 4 5 6 2 5 9	21
5 4 5 1 2 1 2 3 6 3 1 5 4 5 2	impossible