Problem A. Assessments

Input file: `assessment.in`
Output file: `standard output`
Balloon Color: `Yellow`

As the World Cup approaches, FIFA is holding a worldwide assessment of players. In each assessment, we have a line of N distinct players, the players are standing in the order $0, 1, \cdots, N-1$. The assessment is carried out in k rounds, in each round the players run for a long time. When they finish the round, it sometimes happens (with probability p) that exactly one pair of adjacent players end up swapping their positions; this pair of adjacent players occurs randomly with equal chance. FIFA wants your help finding out the probability that the final state after the k rounds will have the players x and y swapped.

Input

The first line of the input contains a single integer $1 \le T \le 100$ the number of test cases. Each test case consists of one line that contains 5 space separated numbers N, p, x, y, k; where $1 \le N \le 50$ is the number of players, $0 \le x, y < N$ are the numbers of the players of the query, $0 \le k \le 3000$ is the number of rounds of the assessment, and p is a floating point number $0.0 \le p \le 1.0$.

Output

For each test case, output a single line displaying the case number and the answer probability rounded to exactly 5 decimal places.

Example

assessment.in	standard output
3	Case 1: 0.33333
4 1.0 1 1 2	Case 2: 0.00000
4 0.0 0 2 1	Case 3: 0.50000
2 0.5 0 1 1	

Problem B. Breaking the Curse

Input file:	`curse.in`
Output file:	`standard output`
Balloon Color:	`Violet`

Egypt has finally qualified to the World Cup 2018 after 28 years. It was almost a curse that has been broken. The main factor of this curse was the goal of Abdelghany in the World Cup 1990. Abdelghany has been talking over and over again about this goal that Egyptians started to regret that they have even qualified to that World Cup. Now, since Egypt has made it to the World Cup, lots have said that the curse is over, but of course Abdelghany disagrees. The curse is about someone scoring for Egypt there after him, not only participating. Hence, all Egyptians are praying day and night for any player to score in the 2018 World Cup.

Of course all the hopes are for Egypt's star Salah to score in the World Cup. Egyptians are interested in knowing the probability that Salah will score in the World Cup and end this curse. Egyptian Scientists decided to calculate this probability for the benefit of the country and the sanity of its people. They encapsulated Salah's traits, like DNA, personality and quality in a string s_1 and did the same for Abdelghany in a string s_2. Now, they will study the similarity of these two strings in order to calculate the probability of Salah scoring in the World Cup, like Abdelghany. They are making some experiments initially in order to be sure and they want your help, by answering q queries each consisting of two integers L and R; in each query you should find the number of intervals $[i, j]$ (inclusive) where $L \leq i \leq j \leq R$ and $s_1[i \cdots j]$ is a substring of s_2.

Input

The first line of the input contains a single integer $1 \leq T \leq 50$ the number of test cases. Each test case begins with two lines, the first line is s_1 and the second line is s_2. The two lines are followed by a line containing a single integer q the number of queries, then followed by q lines each containing two space separated integers L, R the values of the query; where the strings s_1, s_2 consist of lower case English characters, $1 \leq |s_1|, |s_2|, q \leq 75 \cdot 10^3$.

Output

For each test case output a line displaying the case number, followed by q lines each containing a single integer which is the answer to the corresponding query.

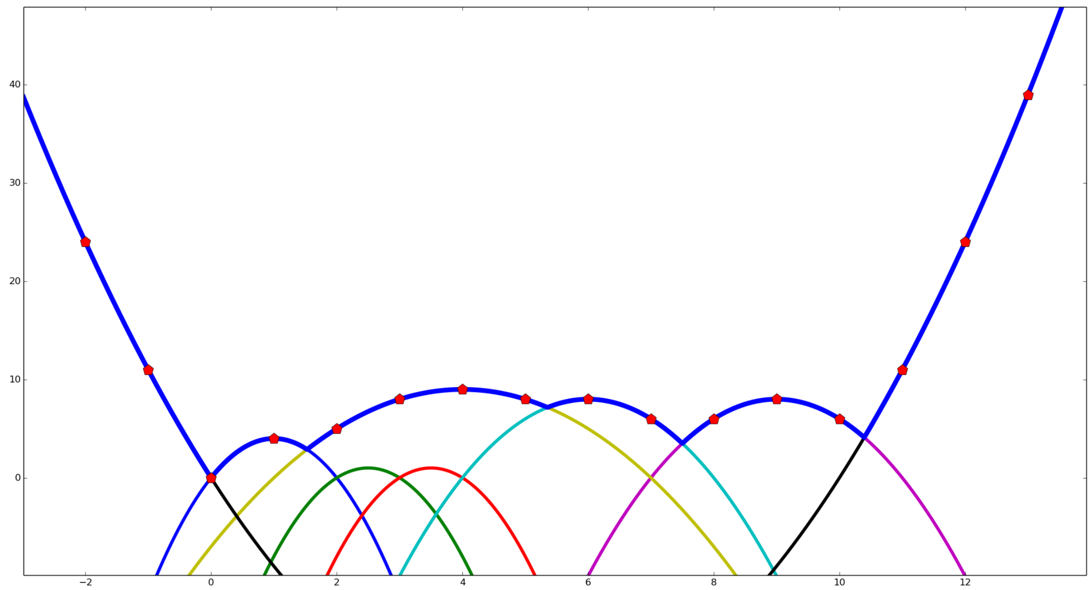
Example

<table>
<tr><th>curse.in</th><th>standard output</th></tr>
<tr><td><pre>2
bbc
abb
4
1 3
1 2
2 3
3 3
hadafkaselalaam
kas
1
1 10</pre></td><td><pre>Case 1:
3
3
1
0
Case 2:
8</pre></td></tr>
</table>

Problem C. Cheering Parabolas

Input file:	`parabolas.in`
Output file:	`standard output`
Balloon Color:	`Blue`

The World Cup is approaching, and a lot of people will be sitting to cheer for their teams. There are N cheering groups for N different teams. Since we are looking at the cheering area from a far place, it looks like a 2d plane of rows and columns. The value on the x axis is the column number, which lies between $[-10^6, 10^6]$. The value on the y axis is the row number. In this plane, each cheering group will sit in the shape of a parabola on the form $y_i(x) = A_i x^2 + B_i x + C_i$, where $y_i(x)$ is the row of the $i-$th cheering group at column x. As it is well known that the team of highest cheering group wins, we would like to know at a specific column x, what is the row of the highest cheering group in this column (i.e. the maximum $y_i(x)$ for all i). Since this is a glorious stadium, you have to answer Q queries about that to check a variety of columns.



Input

The first line contains a single integer $1 \le T \le 100$ the number of test cases. Each test case starts with a line containing a single integer $1 \le N \le 1500$ the number of cheering groups. The following N lines each has 3 integers A_i, B_i, C_i (where $-10^5 \le A_i, B_i, C_i \le 10^5$). The queries start by a line containing a single integer $1 \le Q \le 5 \cdot 10^4$ the number of queries. The following Q lines each contains a single integer x the column number, where $-10^5 \le x \le 10^5$.

Output

In the beginning of each test case, print a single line displaying the case number, followed by Q lines each containing 2 space separated integers; the first integer is the index i (0-based) of the cheering group (since we need to know who they are cheering for, if there are multiple such groups, just print the index of any of them), and the second is the row number $y_i(x)$ of the group i at column x.

Example

parabolas.in	standard output
1	Case 1:
7	6 11
-4 8 0	6 24
-4 20 -24	6 0
-4 28 -48	0 4
-2 24 -64	5 5
-2 36 -154	5 8
-1 8 -7	5 9
1 -10 0	5 8
16	3 8
-1	3 6
-2	4 6
0	4 8
1	4 6
2	6 11
3	6 24
4	6 39
5	
6	
7	
8	
9	
10	
11	
12	
13	

Problem D. Dream Team

Input file:	`dream.in`
Output file:	`standard output`
Balloon Color:	`Pink`

Before the World Cup, we would like to form the ultimate dream team. The team that will win against all teams. We have a big group of the top N players in the world, where each player has a number representing his skill level. Two players i and j with skill levels a_i and a_j have a compatibility of passing the ball between each other; this compatibility level is measured by the greatest common divisor of their skill levels (i.e. $compatibility(i, j) = gcd(a_i, a_j)$). We would like to decide a strategy that connects all the players of the dream team with a tree of connections between the players. If two players are directly connected in the chosen strategy, then they will pass the ball between each others during the matches, with the compatibility level between them. The compatibility of the strategy is the sum of the compatibility levels of its connections. What is the maximum total compatibility of the chosen strategy that connects all players?

Input

The first line of the input contains a single integer $1 \leq T \leq 50$ the number of test cases. Each test case consists of one line that contains $N + 1$ space separated integers; the first integer is N, followed by the skill levels $a_1, \cdots, a_n$; where $1 \leq N \leq 10^5$ and $1 \leq a_i \leq 10^5$ for all i.

Output

For each test case, print a single line displaying the case number and the maximum compatibility of the dream team's strategy.

Example

dream.in	standard output
2 2 3 6 5 5 6 7 10 21	Case 1: 3 Case 2: 17

Note

In the second test case we connect the players $3-5$ with compatibility 7, players $1-4$ with compatibility 5, players $2-5$ with compatibility 3 and players $2-4$ with compatibility 2; therefore the total compatibility of the dream team's strategy is 17.

Problem E. Evaluations

Input file: `evaluations.in`
Output file: `standard output`
Balloon Color: `Gold`

For World Cup marketing purposes, FIFA wants to evaluate the compatibility of different players. Some players have connections that affect their compatibility scores, like being from the same country or playing in the same team. These connections are well studied, and for some pairs of players, an integer value is assigned based on these connections. After such studies, FIFA found out that the set of players with connections form an undirected tree. In more details, players are the nodes of the tree and edges connect players with a connection such that the edge's weight is equal to the value of this connection. It is guaranteed that these edges will form a tree. The compatibility score of players u, v is:

- If there is an edge (u, v) in the tree, then it is the weight of that edge.
- Otherwise, it is the weight of the path between u and v, such that the weight of a path is the product of the weights of its edges.

Given the tree of players along with edges' weights, FIFA is interested in counting the number of distinct unordered pairs of players u, v such that their compatibility score is a product of exactly two distinct primes.

Input

The first line of the input contains a single integer $1 \leq T \leq 100$ the number of test cases. Each test case consists of n lines, the first line containing a single integer n. Each of the remaining $n-1$ lines consists of 3 space separated integers x y w; each of those denotes an edge between the nodes x and y with weight w; where $1 \leq x, y \leq n \leq 10^5$ and $1 \leq w \leq 10^5$.

Output

For each test case output a single line displaying the case number and the count.

Example

evaluations.in	standard output
1 5 1 2 2 2 3 1 1 4 3 1 5 6	Case 1: 3

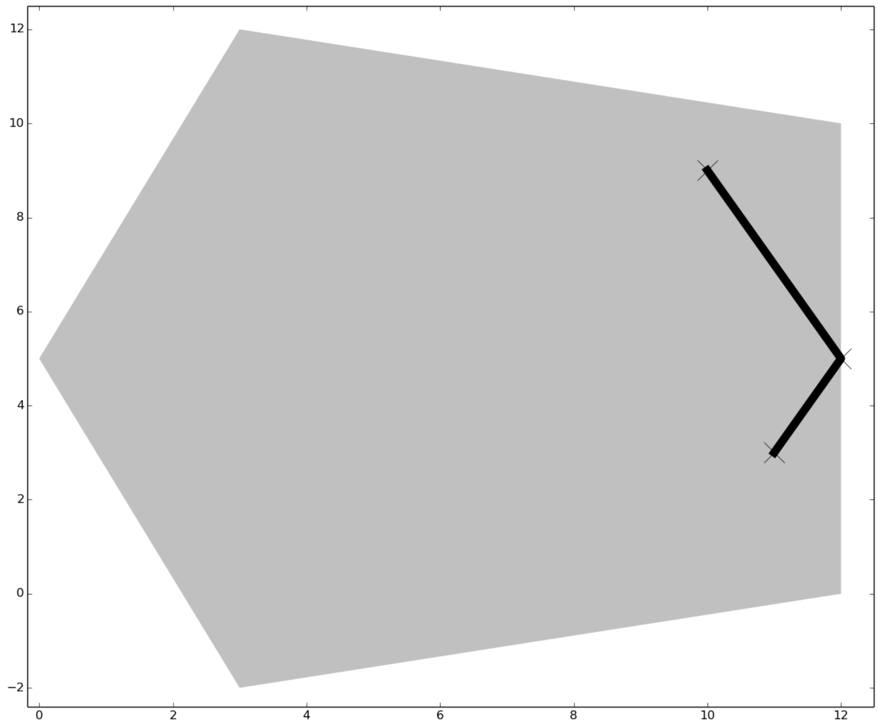
Note

An unordered pair is a set of two distinct elements. $\{a, b\} = \{b, a\}$.

Problem F. Forgot the Flag!

Input file:	`flags.in`
Output file:	`standard output`
Balloon Color:	`Green`

After walking through a lot of people, in order to cheer for the Egyptian team in the World Cup. You finally found your cheering spot in the field of the stadium. The field is shaped like a convex polygon. Then you suddenly remembered that you forgot the Egyptian flag. Since you want to cheer so hard for Egypt, you had to get the flag from any shop on the borders of the stadium (There is a shop on every point on the borders of the stadium), then get back to your spot. Sadly someone took your spot, but since you always have a backup plan, you will go to the second spot you booked. Since all the borders are filled with shops, which shop should you go to in order to minimize the total distance you have to walk from your first spot, to the shop, then to your second spot? You need to answer this question Q times since it happened with a lot of your friends.



Input

The first line of the input consists of a single integer $1 \leq T \leq 50$ the number of test cases. Each test case starts with a line containing a single integer n the number of vertices of the stadium. The following n lines each contains 2 space separated integers x_i, y_i the coordinates of the i-th vertex of the stadium. There are no 3 consequent vertices of the polygon that are collinear. The remaining input of the case starts by a line containing a single integer Q. The following Q lines, each containing 4 space separated integers x_1, y_1, x_2, y_2, where (x_1, y_1) is your first spot and (x_2, y_2) is the location of your second spot (both not outside the stadium of course). The vertices are given in anti-clockwise order or clockwise order, where $3 \leq n \leq 5 \cdot 10^4$, $1 \leq Q \leq 10$ and all coordinates $|x|, |y| \leq 10^9$.

Output

For each test case output a single line, displaying the case number followed by Q lines each containing 3 space separated decimal numbers rounded to exactly 7 decimal places. The 3 numbers will be on the format 'd x y', where d is the shortest distance you will walk in total and (x, y) is the location of the shop on the borders, from which you will pick up the Egyptian flag. If there are multiple shops, output any of them. The output will be checked with a relative error.

Example

flags.in	standard output
1 5 0 5 3 -2 12 0 12 10 3 12 1 10 9 11 3	Case 1: 6.7082039 12.0000000 5.0000000

Note

The first and second spots are not necessarily distinct.

Problem G. Glorious Stadium

Input file:	`glorious.in`
Output file:	`standard output`
Balloon Color:	`Orange`

A lot of people want to attend the World Cup, so we would like to construct a glorious stadium using the following algorithm:

1. Construct 1st layer of the stadium by building a fence shaped like a regular polygon P_1 with K sides and its in-circle C_1 (with radius R) is the field of the stadium.

2. Construct 2nd layer of the stadium by building a fence shaped like a regular polygon P_2 with K sides and its in-circle C_2, such that C_2 is the circumcircle of P_1, so it surrounds the layer 1.

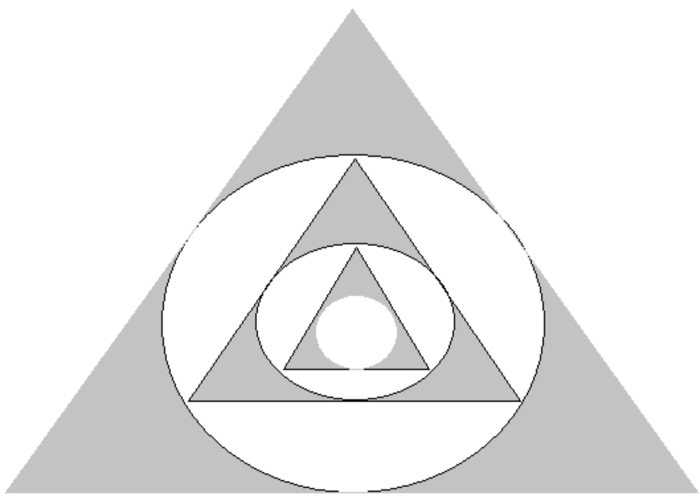
3. Construct 3rd layer of the stadium by building a fence shaped like a regular polygon P_3 with K sides and its in-circle C_3, such that C_3 is the circumcircle of P_2, so it surrounds the layer 2.

   Until $\cdots$

N. Construct $n-$th layer of the stadium by building a fence shaped like a regular polygon P_N with K sides and its in-circle C_N, such that C_N is the circumcircle of P_{N-1}, so it surrounds the layer $N-1$.

For each layer of the stadium, the area of the polygon that is not included in its in-circle tells us the amount of the people who can stay to cheer for the teams (because this is the only place they can sit). We would like to calculate the sum of cheerers areas of all the N layers of the glorious stadium, that is:

$$\sum_{i=1}^{N} Area(P_i \setminus C_i)$$



Input

The first line of the input consists of a single integer $1 \leq T \leq 10^5$ the number of test cases. Each test case consists of one line containing 3 space separated integers N, R, K. N is the number of layers of the stadium, R is the radius of the stadium C_1 and K is the number of sides of the polygons; where $1 \leq N \leq 10^5, 1 \leq R \leq 100, 3 \leq K \leq 2 \cdot 10^4$ and $5 \cdot K \geq N$.

Output

For each test case output a single line displaying the case number and the area of the cheerers rounded to exactly 5 decimal places. The output will be checked with a relative error.

Example

glorious.in	standard output
3 3 10 3 2 10 4 5 100 6	Case 1: 4314.57552 Case 2: 257.52220 Case 3: 31096.23444

Note

- in-circle is the largest circle the will fit inside a polygon that touches every side.
- circumcircle is the circle that passes through each vertex of the regular polygon.
- The given figure is a demonstration of the first test case, where the radius of the smallest circle is the given R, and the answer is the area of the grey parts.
- The formula $P \setminus C$ denotes the area covered in P and not covered in C, as shown in the figure.

Problem H. Half Nice Years

Input file:	`halfnice.in`
Output file:	`standard output`
Balloon Color:	`Black`

Salah is really optimistic about the qualification of the Egyptian football team for the world cup 2018. He was thinking deeply about why it didn't happen often that the Egyptian team qualified to the world cup, so he decided to call 2018 a half-nice year. He comes up with the hypothesis that the Egyptian team will always qualify to the world cup only if it is held in a half-nice year. In order to test his hypothesis we need to find the largest half-nice number in a given range $[A, B]$. He says a given number X is half-nice if its decimal representation can be divided into two non-zero halves(of the same length) and the two halves have a non-trivial common divisor (there's a number $d > 1$ that divides both parts). In case a number X have odd number of digits $(2r + 1)$, then the first half will have the first r digits in the decimal representation of X (from the left), and the second half will contain the remaining $r + 1$ digits in the decimal representation of X.

Input

The first line of the input consists of a single integer $1 \leq T \leq 10^5$ the number of test cases. Each test case consists of one line containing 2 space separated integers A, B, where $10 \leq A \leq B \leq 10^{13}$.

Output

For each test case output a single line displaying the case number, then followed by either a single integer X, which is the largest half-nice number in the range $[A, B]$ or 'impossible' if there are no such numbers.

Example

halfnice.in	standard output
3 20 21 10 2019 4000 4005	Case 1: impossible Case 2: 2018 Case 3: 4005

Note

- The number 20122 is a half-nice number (hopefully the Egyptian team qualifies that year), because it's two halves 20 and 122 have a non-trivial common divisor which is 2.
- The number 2000 is not a half-nice number because the second half is zero.
- In the third, test case 4005 is half-nice because 40 and 5 share the non-trivial common divisor 5. It is allowed for the second half to have trailing zeros.

Problem I. Important matches

Input file:	`important.in`
Output file:	`standard output`
Balloon Color:	`light blue`

In the qualification rounds of the World Cup, there were two disjoint sets of teams $S_1 = [1, \cdots, N]$ and $S_2 = [1, \cdots, M]$. Given a matrix X that decides the importance of the match between team $i \in S_1$ against team $j \in S_2$ by the value $X_{i,j}$. S_1 represents the rows of the matrix X, and S_2 represents the columns of the matrix X. We would like to answer Q queries, where each decides the importance of all the matches played by all the teams from the subgroup $[A, \cdots, C]$ from S_1 against all teams from the subgroup $[B, \cdots, D]$ from S_2, such that $1 \leq A \leq C \leq N$, $1 \leq B \leq D \leq M$. The importance of a set of matches between two subgroups is determined by the median importance of all the matches between the teams from the first subgroup against all the teams from the second group, namely the median of the submatrix $X_{A \cdots C, B \cdots D}$.

Input

The first line of the input contains a single integer $1 \leq T \leq 100$ the number of test cases. Each test case begins with a line containing 3 spaces separated integers N, M, Q. The importance matrix X is given by the next N lines each containing M space separated integers. The remaining Q lines, each contains 4 space separated integers A, B, C, D of the query; where $1 \leq N, M \leq 200$, $1 \leq Q \leq 10000$, and for the given importance matrix X, $1 \leq X_{i,j} \leq 2000$ for all i, j.

Output

For each test case output a line displaying the case number, followed by Q lines each containing a single integer, the importance of the matches of the corresponding query.

Example

<table>
<tr><th>important.in</th><th>standard output</th></tr>
<tr><td><pre>1
3 4 2
1 2 3 1
2 1 1 4
7 8 9 3
1 1 1 1
1 2 3 4</pre></td><td><pre>Case 1:
1
3</pre></td></tr>
</table>

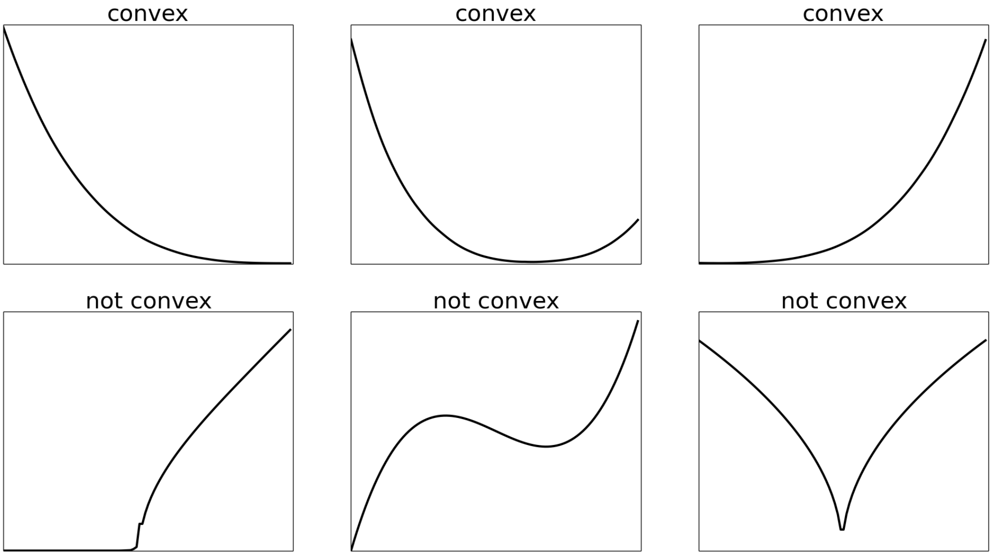
Note

- The median of a set of numbers is the middle element in the sorted list of the given set. If the set has two middle elements, then we choose the second one. For example the median of {2, 1, 3} is 2 and the median of {4, 2, 3, 1} is 3.
- The second query asks for the median of the submatrix that is excluding the first column. This submatrix contains the elements {2, 3, 1, 1, 1, 4, 8, 9, 3}, which has the sorted form {1, 1, 1, 2, 3, 3, 4, 8, 9}, therefore the median is 3.

Problem J. Jacked Tickets

Input file: `jacking.in`
Output file: `standard output`
Balloon Color: `Purple`

The World Cup is coming soon, everyone is interested in getting tickets, so it became really expensive to get any ticket. There are a lot of thieves who became interested in stealing the tickets of the matches. We would like to reduce the loss of the situation. We decided to partition the n tickets that we have into m non-empty groups, because we have m thieves, who will ruin the match if any of them doesn't get any tickets. Each thief will get exactly one group. This partitioning is done in order to minimize the total loss caused by stealing the n tickets. The total loss of stealing the n tickets is the sum of the loss caused by each thief, given that a thief who gets k tickets will cause a loss of $P(k)$ (for a given convex function P). Given the values $P(1), P(2), \cdots, P(n)$, You need to output the minimum total loss caused by the thieves.



Input

The first line of input contains a single integer $1 \leq T \leq 150$ the number of test cases. Each test case begins with a line containing $N+1$ space separated integers, the first is the maximum number of tickets N followed by N integers, the values of $P(1), P(2), \cdots, P(N)$. The queries then begin with a line containing a single integer Q the number of queries, followed by Q lines each containing two space separated integers n, m; where $1 \leq n, m \leq N \leq 500$, $1 \leq Q \leq 10^4$ and $0 < P(i) \leq 10^{15}$ for all i.

Output

For each test case output a single line displaying the case number, then followed by Q lines, each containing a single integer which is the answer of the corresponding query n, m. The answer to a query n, m is namely the minimum total loss caused by stealing n tickets by m thieves or 'impossible' if it's impossible to save the match from being ruined by some of the thieves.

Example

jacking.in	standard output
2 3 1 1 1 2 3 1 2 1 6 15 12 10 13 19 28 5 1 1 1 4 4 3 4 4 6 6	Case 1: 1 1 Case 2: 15 impossible 42 60 90

Note

- A function f is convex if and only if for all variables x_0, x_1, x_2 such that $x_0 \le x_1 \le x_2$ the property $(x_2 - x_1)f(x_0) + (x_1 - x_0)f(x_2) \ge (x_2 - x_0)f(x_1)$ is satisfied, or also equivalently $\lambda f(x_1) + (1-\lambda)f(x_2) \ge f(\lambda x_1 + (1-\lambda)x_2)$ is satisfied for $0 \le \lambda \le 1$.
- If P and Q are convex functions, then $P+Q$ is also convex, where $(P+Q)(x) = P(x) + Q(x)$.

Problem K. Katryoshka

Input file:	`katryoshka.in`
Output file:	`standard output`
Balloon Color:	`White`

The Egyptian football team will be in Russia for the World Cup, of course they all would like to buy souvenirs for their families. Luckily they met the king of souvenirs of the famous Russian souvenir Matryoshka; the king makes his masterpiece Katryoshka using different kinds of pieces: wooden eyes, wooden mouths, and wooden bodies. He can form a nice Katryoshka using one of the following ways:

- Two eyes and a body.
- Two eyes, mouth and a body.
- Eye, mouth and a body.

If the king has n eyes, m mouths and k bodies, what is the largest number of Katryoshkas that can be made by the king?

Input

The first line of the input contains a single integer $1 \leq T \leq 100$ the number of test cases. Each test case consists of a single line containing 3 space separated integers 'n m k' the number of eyes, mouths and bodies respectively; where $0 \leq n, m, k \leq 10^8$.

Output

For each test case output a line displaying the case number and a single integer which is the largest number of nice Katryoshka pieces.

Example

katryoshka.in	standard output
4 1 2 3 0 11 2 14 21 23 90 24 89	Case 1: 1 Case 2: 0 Case 3: 14 Case 4: 57

Problem L. Lazy ERCD

Input file:	`lazy.in`
Output file:	`standard output`
Balloon Color:	`Red`

FIFA changed the style of the World Cup. There are no group rounds anymore, therefore the World Cup will be a pure knockout competition. In a knockout competition, each match that is played between two teams, the losing team is knocked-out of the competition and will not play again, until there's exactly one winner (we are not concerned about other positions, only the first place).

Having N teams in the World Cup, our ERCD wants to know how many matches he needs to watch (he will really watch all matches). He is lazy to count that number, so he needs you to write a program that calculates it.

Input

The first line of the input contains a single integer $1 \leq T \leq 100$ the number of test cases. Each test case consists of 1 line, containing a single integer N, the number of teams; where $1 \leq N \leq 100$.

Output

For each test case output a single line displaying the case number, followed by the number of matches the ERCD will watch.

Example

lazy.in	standard output
3 2 87 4	Case 1: 1 Case 2: 86 Case 3: 3