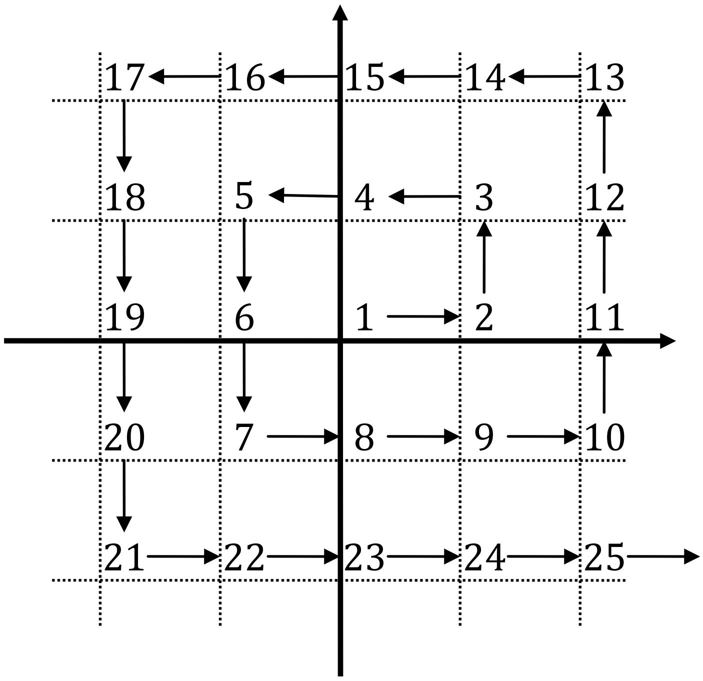
Problem A – Spiral

Little Long is a genius kid and loves special structural thing. These days, he is interested in spiral structure. He drew a Descartes plane and started to fill numbers at integer coordinates. He started at the origin of the plane and continuously filled numbers in counter-clockwise direction. The first 25 numbers are filled like the following figure.



He realized that he can not continue this process for very long because he was going out of paper very soon. Oh, déjà vu, he seemed to be the next and greater Fermat. He wondered which number would be filled at coordinate (x, y). You should help him to figure it out.

Input

The first line of the input is a positive integer T denotes the number of test cases. Then T test cases follow. Each test case has two numbers x, y. The absolute values of all the numbers in the input do not exceed 100.

Output

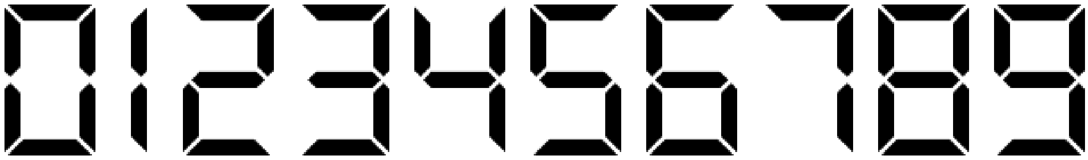
For each test case, print the number filled at coordinate (x, y).

Sample

Sample input	Output for sample input
3 0 0 -1 -1 1 -2	1 7 24

Problem B – Plastic sticks

Little Long loves playing with his plastic sticks. He usually uses his sticks to form digits like the following picture.



In his collection, there are N sticks and he can create many different numbers. Little Long never creates numbers with unnecessary leading zeros. (i.e He never uses 14 sticks to create 010). He wants to know what are the maximal and the minimal non negative number he can create using all of his sticks. Your task is to help him calculate these numbers.

Input

The first line of input is the number T – the number of test cases. Then T test cases follow. Each test case is written in one line consist of a positive integer N - the number of plastic sticks in the collection. ($N \leq 1000$)

Output

For each test case, you should print in two lines the minimal number and the maximal number he can create. If it is impossible to create the number, print -1 instead.

Sample

Sample input	Output for sample input
2 2 4	1 1 4 11

Problem C – Queens

As little Long grows up and enters the second grade, he starts to interest in chess. In this game, the queen is the most powerful piece, able to move any number of unoccupied squares vertically, horizontally, or diagonally. At the same time, a queen is able to capture an enemy by occupying the square on which an enemy piece sits, as long as there is no other piece between the queen and the captured piece.

In other words, if a queen is placed in cell (X, Y) (X-th row from the top, Y-th column from the left), it can moves to (or captures the piece in the cell) the following cells: $(X, Y + k)$, $(X, Y - k)$, $(X + k, Y)$, $(X - k, Y)$, $(X + k, Y + k)$, $(X + k, Y - k)$, $(X - k, Y + k)$, $(X - k, Y - k)$ as long as there is no other piece between the queen's position and the destination, exclusively. Obviously, the queen cannot move outside the board.

A cell is called safe if it is empty and we can put an enemy piece there and there is no queen can capture it in 1 move.

Given a chessboard of m rows and n columns ($1 \le m, n \le 100$) and k queens on the board, your task is to count the number of safe cells.

Input

The first line contains an integer T ($T \le 50$), the number of test data.

T blocks of test data follow, each starts with a line contains three integers m, n, k. Then k lines follow, each line contains two numbers X, Y which are the positions of a queen. All the queens are placed in different cells.

Output

T lines, each line contains one number which is the answer of a test data.

Sample

Sample input	Output for sample input
2 4 4 1 2 2 4 4 2 2 2 3 3	4 2

Problem D – Pythagorean triple

Little Long is now in grade 7th and he shows great ability and passionate in geometry. After having learned about right triangle and Pythagorean Theorem, he had researched more on the internet about *Pythagorean triple* and its distribution.

The Pythagorean triple is defined as three positive integers a, b, and c, such that $a^2 + b^2 = c^2$

Figure 1 is the distribution of pairs (a, b) where exists a number c such that (a, b, c) is a Pythagorean triple. Looking at it, little Long realized that it is more compressed than he had thought. Thus, he wants to verify with a set of random integers. Given a set S consists of N positive integers, your task is to help him count the number of different Pythagorean triples in set S. Triple (x, y, z) and triple (y, x, z) are considered the same.

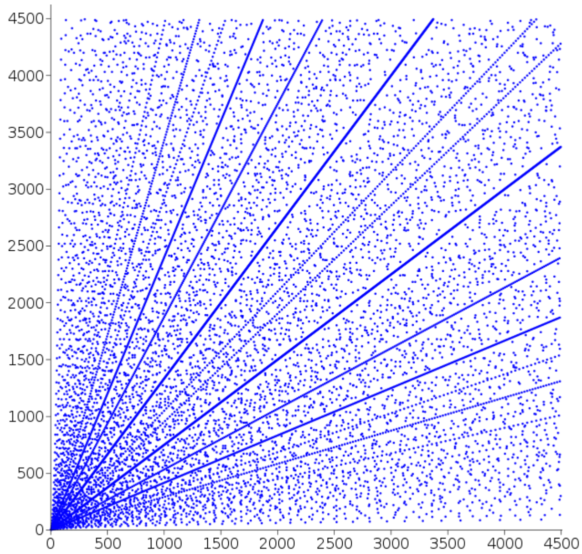


Figure 1

Input

The input starts with the number T $(T \le 20)$ – the number of test cases. Then T test cases follow. Each test cases starts with number N $(N \le 100)$. Following this are N numbers of set S. All the numbers in the set are guaranteed to be unique and do not exceed 10^4.

Output

For each test case, display a single line containing the case number and the number of Pythagorean triples formatted like the sample data.

Sample

Sample input	Output for sample input
2 6 1 2 3 4 5 6 5 13 12 5 4 3	Case #1: 1 Case #2: 2

Problem E – Robot

Little Long is now in grade 10th. He is making a transportation robot to compete in the Intel ISEF. After 3 days of working with his supervisor, Little Long managed to create a robot which can accept four kinds of instruction: $'u', 'd', 'r', 'l'$ – which mean moving one step upward, downward, to the right and to the left respectively.

He is a very careful and strict person so he follows the Software Development Life Cycle. He brings the robot inside a room of $m * n$ squares for testing. He puts the robot at a random square and gives it a sequence of instructions (a string consists of only four characters – u, d, r, l). The robot executes the instruction from left to right and follows the direction of the instruction. The robot will not execute an instruction which asks him to move out of the room, so if there is such instruction, the robot will simply ignore the instruction and move to the next one. Little Long is curious with a question: how many different positions in which the robot can finish its journey (i.e. the initial location can be any position in the room).

Input

The first line of the input contains one integer T ($T \leq 10$), which is the number of cases in this input set. There are T test cases follow. Each test case consists of 2 lines:

- The first line has only 2 numbers m and n ($m, n \leq 10^5$) which is the width (left-right dimension), the length (up-down dimension).
- The second line has one string s ($length(s) \leq 10^5$), which is the instruction sequence.

Output

T lines, which are the answers for T test cases with the same format as the sample data.

Sample

Sample input	Output for sample input
3 2 2 ul 3 3 ulrrrr 4 2 Uddlur	Case 1: 1 Case 2: 2 Case 3: 3

Problem F – String reconstruction

Little Long is now in grade 12th and is selected to represent Vietnam in the International Olympiads in Informatics (IOI). In a special training for the competition, he has just been introduced to the *suffix array* data structure.

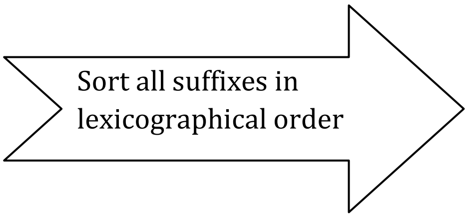
A suffix array is defined as a sorted array of all suffixes of a string. It is a simple, yet powerful data structure which is used in full text indices, data compression algorithms and within the field of bioinformatics.

Let $S = s_1 s_2 s_3 \dots s_n$ be a string and let $S[i,j]$ denote the substring of S ranging from i to j. All $S[i,n]$ with $1 \le i \le n$ are suffixes of S.

The suffix array A of S is now defined to be an array of integers providing the starting point of suffixes of S in lexicographical order. This means, an entry $A[i]$ contains the starting position of the i-th smallest suffix in S and thus, for all $1 < i \le n$: $S[A[i-1],n] < S[A[i],n]$.

Consider S = "BANANA":

i	$S[i,n]$
1	BANANA
2	ANANA
3	NANA
4	ANA
5	NA
6	A



i	Sorted suffixes
6	A
4	ANA
2	ANANA
1	BANANA
5	NA
3	NANA

We got the suffix array A of $S = [6,4,2,1,5,3]$. Little Long is wondering: "Given an suffix array A of S, can we reconstruct string S which consist of only upper-case characters?".

Input

The input consists of T test cases. Each test case starts with an integer N denotes the length of the string (i.e. the number of suffixes). Then N integers denote the suffix array follows. All the numbers in the input are positive integers and do not exceed 50.

Output

For each test case, print in one line the reconstructed string. If there are multiple solutions, print the one that comes first lexicographically. If the string can not be reconstructed from the given suffix array, print "NO SOLUTION" instead.

Sample input	Output for sample input
2 4 4 3 2 1 3 2 3 1	AAAA BAB

Problem G – Bus services

Little Long is now in team number X and currently participating in the ACM National Contest at FPT University campus located in Hoa Lac. It seems that his team is doing very well and will be very likely to be crowned the National Champion. They are planning that after the contest, they will return to the city centre to celebrate their victory. In order to go back to the city centre, they will take the buses to go along "Thang Long Avenue" from Hoa Lac to Hanoi centre.

The distance from Hoa Lac to Hanoi centre is 30 Km and there is a bus stop every 1 Km. Thus, there are 31 bus stops in total, numbered 0 to 30. The bus stop in Hoa Lac is numbered 0 and the bus stop in Hanoi numbered 30.

There are M bus services along the avenue numbered from 0 to $M-1$. We know that the speed of a bus is 60 Km/h (1 Km/min); we also know that the stop time at each bus stop is a negligible amount and could be ignored. To board a bus that leaves a bus stop at time t, they have to present at that bus stop before time t (i.e. time $t-1$ is the latest time that they have to present at the bus stop) or they have to wait for the next bus.

Each bus services do not stop at all the bus stops on its way. It will stop at some scheduled stops only. Each service also has its own starting time and its own frequency.

For example, there could be 4 bus services:

Bus service #0 Stop at 0, 6, 12, 18, 24 Start at time 10, every 60 minutes					
Bus stop	0	6	12	18	24
1st bus	10	16	22	28	34
2nd bus	70	76	82	88	94
3rd bus	130	136	142	148	154

Bus service #1: Stop at 16, 20, 24, 30 Start at time 26, every 120 minutes				
Bus stop	16	20	24	30
1st bus	26	30	34	40
2nd bus	146	150	154	160
3rd bus	266	270	274	280

Bus service #2: Stop at 15, 30 Start at time 68, every 120 minutes		
Bus stop	15	30
1st bus	68	83
2nd bus	188	203
3rd bus	308	313

Bus service #3: Stop at 25, 26, 27, 28, 29, 30 Start at time 30, every 60 minutes						
Bus stop	25	26	27	28	29	30
1st bus	30	31	32	33	34	35
2nd bus	90	91	92	93	94	95
3rd bus	120	121	122	123	124	125

These tables only contain the first 3 buses of each service in a day. Following buses will continue as scheduled and day change does not effect. It is possible that they have to walk along the avenue and their speed is 4 Km/h (1 Km/15 min). Please note that these bus services are **one-direction** because the returned buses go by a different route. With these services and the current time is 0. To travel from bus stop 0 to bus stop 30, they can have several options:

- At time 0, they start to walk 30 Kms. they will reach the destination in 450 minutes.
- Wait 10 minutes, take the service #0. Get off at bus stop 24 at time 34. Now, they can not take the first bus of service #1 because it will leave at time 34. They have to wait for 120

minutes and the second bus of service #1 and it will arrive at the destination at time 160.

- Wait 10 minutes, take the service #0. Get off at bus stop 24 at time 34 and walk to the destination. They will arrive at $34 + 6 \times 15 = 124$.
- Wait 10 minutes, take the service #0. Get off at bus stop 24 at time 34. Walk to bus stop 25. They will arrive at bus stop 25 at $34 + 1 \times 15 = 49$. They have to wait until time 90 to take the second bus of service #3. They will arrive at the destination at time 95.
- Wait 10 minutes, take the service #0. Get off at bus stop 12 at time 22. Walk to bus stop 15. They will arrive at bus stop 15 at $22 + 3 \times 15 = 67$, just on time for them to catch the first bus of service #2 leaving bus stop 15 at time 68. They will arrive at the destination at time 83.

Your task is to find the optimal option for each of these purposes:

- The earliest arrival time.
- The earliest arrival time and the walking distance does not exceed X Km.
- The earliest arrival time and the number of bus taken does not exceed Y.
- The earliest arrival time, the walking distance does not exceed X Km and the number of bus taken does not exceed Y.

Input

The first line of the input contains number T ($T \leq 100$). Then, T test cases follow:

- The first line contains number M, X, Y ($0 \leq M \leq 10, 0 \leq X \leq 30, 0 \leq Y \leq M$).
- Each of the next M lines contains information of each bus service: start time of first bus S, frequency F. Followed by N numbers describe N stops. These numbers are listed in increasing order. ($1 \leq S \leq 1000, 1 \leq F \leq 1000$)

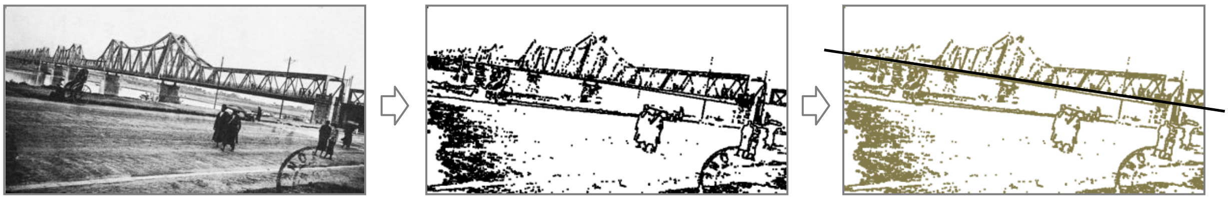
Output

For each test case, you should print the earliest arrival time of 4 options (in minutes). If there is no option, print -1 instead.

Sample

Sample input	Output for sample input
2 4 1 1 10 60 0 5 12 18 24 26 120 16 20 24 30 68 120 15 30 30 60 25 26 27 28 29 30 1 1 1 1 1 1 29	83 95 124 -1 59 -1 59 -1

Problem H – Photo Leveling



Little Long is now pursuing his Ph.D. degree at a distinguished university on the field of image processing. He joins a team to develop an automatic rotation function for a photo editing app. One potential solution is finding the dominant line in the photo, and then aligning it horizontal or vertical. Typically, this dominant line is a straight edge of an object such as the skyline or the edges of buildings. The team has successfully implemented an edge detection algorithm which can output a set of candidate pixels that potentially belongs to the edges of objects. His task is finding the maximum subset of collinear pixels, which are on the dominant line. In addition, he only need to implement a solution for the first testing phase; where there are always at least 60 percent of the input pixels on the dominant line.

Input

The first line contains an integer number K denotes the number of test cases. Following lines are in K groups where each group describes one test case in some lines as follow:

- First line in the group contains the number of pixels $5 \leq N \leq 100000$,
- Each in the following N lines contains two coordinates x_i, y_i of the i^th-pixel. These coordinates are integer numbers in range of -10^9 to 10^9.

Output

For each test case in the input, you need to write two lines to the output:

- The first line should contain the maximum number of collinear pixels M,
- The second line should contain M indices of collinear pixels. The indices should be printed in ascending order with exactly one space in between two consecutive numbers. For consistency, you should also use the one-based numbering, i.e. the index of input pixels starts from 1.

Sample

Sample input	Output for sample input
1 8 0 1 7 4 -1 0 1 1 -9 -4 -7 -3 1 10 6 6	5 2 3 4 5 6

Problem I – Travelling the Mars

After finish his Ph.D in image processing, Little Long joins the National Aeronautics and Space Administration (NASA) to work on a very challenging project. He is in a group with other scientists to collect information about the Mars.

Scientists from NASA commonly use the *ring-strip coordinate system* when exploring the Mars. This system is defined as follows:

- The Mars is assuming to be a completely perfect sphere with radius of 100.
- The surface of the Mars is imaginarily divided using M rings (labeled $0, 1, 2, \dots, M-1$) and N strips (labeled $0, 1, 2, \dots, N-1$) as illustrated in the next figure. The intersection of a strip and a ring is called a *zone*. In total, there are $M \times N$ zones.
- All strips are identical in shapes.
- M rings divide edges of any strip into M equal arcs (i.e. any line of longitude travels the same distance in each ring).

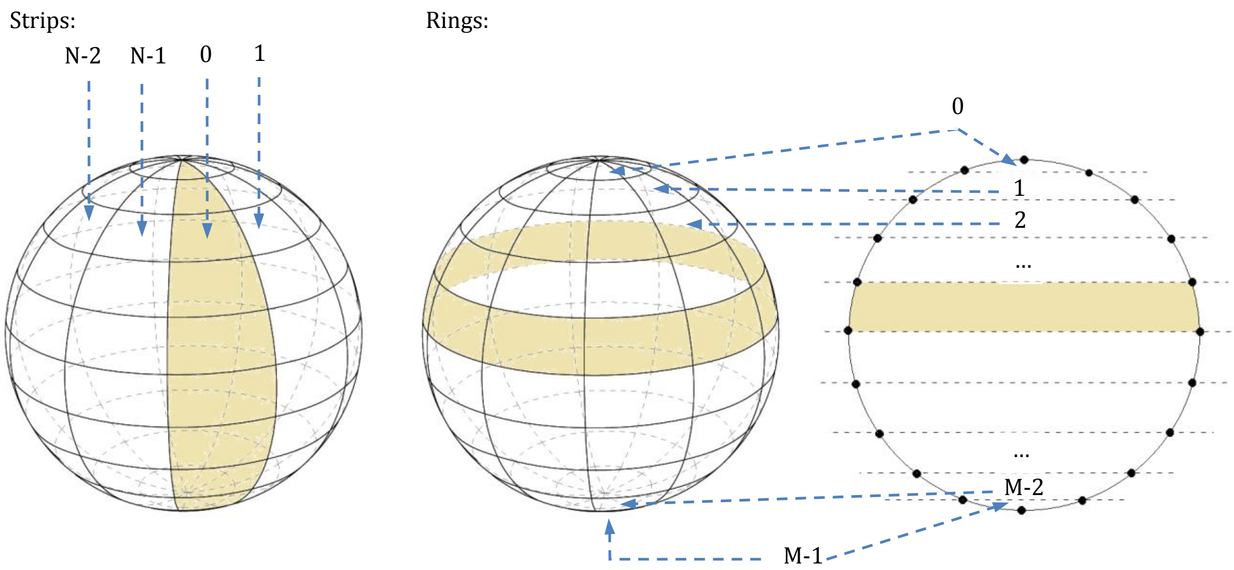


Figure 2: Illustration of *strip* and *ring*.

NASA scientists are developing a new-generation robot for exploring the Mars. The robot is programmed to perform the following routine:

- Landing on the Mars at zone A (the entry zone).
- Robot plans a route to zone B (the return zone). A route is defined as a sequence of **adjacent zones** which starts at A and ends at B. Two zones are adjacent if they have at least one point in common.
- Robot follows the route to navigate from A to B. Every time passing by a zone (included A and B), robot captures one image of the entire zone and send it back to the Earth as a radio signal. The size of the image is naturally proportional to the surface area of the zone. We call the total surface area of all zones in the route **the surface area of the route.**
- After sending the last photo, robot returns to the Earth.

Because of the long distance between the Earth and the Mars, radio signals sent back from robot will most likely be degraded along the way. Therefore, it is a rule of thumb to avoid transmitting too many or too lengthy signals. NASA scientists agree that at step 2, robot should plan to minimize the surface area of the route.

Input

There will be a number of lines each one which contains 6 numbers: M, N, r_A, s_A, r_B, and s_B describing one test case. M is the number of rings. N is the number of strips. r_A and s_A is the ring index and the strip index of point A. r_B and s_B is the ring index and the strip index of point B. The input satisfies $2 \leq M,\ N \leq 100$; $0 \leq r_A,\ r_B \leq M-1$; $0 \leq s_A,\ s_B \leq N-1$. There are no more than 250 test cases.

Output

Output should consist of the same number of lines with that of the input. Line i-th contains one real number which is the minimum surface area for test case i-th printed to exactly 3 decimal digits.

Sample

Sample input	Output for sample input
`2 2 0 0 1 1` `8 9 0 3 7 3`	`62831.853` `13962.634`

Problem J – Exploring the Mars

Now, the first phrase of the exploring the Mars was finished, little Long and his colleagues have their general ideas about every zones of the Mars. They believe that in there is a very high chance of finding water on a specific zone. Thus, in the next phrase of this mission, they will use an upgraded robot to search only in this zone. It will follow a routine:

1. Place in the 1st position, mark this position as known,
2. Number of known position $n = 1$,
3. Repeat operations:
    a) From a known position u, go to a new position $n + 1$,
    b) Mark this position as known and increase n by 1,
    c) Find the trace of water,
    d) Recalculating the map of known positions.

The recalculating the map of known positions is a very important task because the rocky surface may force the robot to deviate from its planed route. In this process, one important parameter of the map of known positions is the longest distance between two known positions which shows how breath the search is.

Little Long is responsible for doing the recalculating the map function. Because this function is very important, he wants somebody to write the same function so that they can crosscheck with each other. Can you help him to do that?

Input

The input starts with a number T ($T \leq 10$) which denotes the number of test cases. Then T test cases follow. Each test case starts with a number Q ($1 \leq Q \leq 30000$) which denotes the number of repeat operations. Then Q lines follow, each consist of 2 positive integers u and c where u is the known position (step 3a) and c is the distance to the new position. ($1 \leq c \leq 100000$)

Output

For each repeat operation in the input, print the longest distance between two known positions.

Sample

Sample input	Output for sample input
1 5 1 10 1 10 2 5 3 5 5 10	10 20 25 30 40