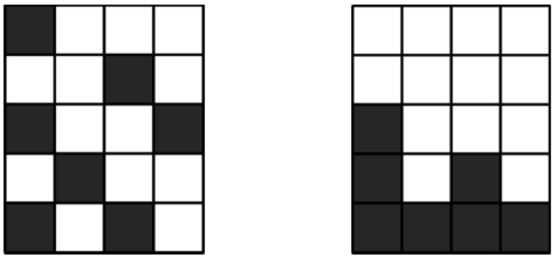
The 38th Annual
ACM International Collegiate
Programming Contest
Asia Regional - Daejeon

Problem A
Boxes
Time Limit: 1 second

A grid with $\boldsymbol{m}$ rows and $\boldsymbol{n}$ columns is given. Some cells of the grid are occupied by boxes. If all the boxes of the grid move downwards until they cannot move any further, the boxes will be stacked at the bottom of the grid. An example is shown in Figure 1. In Figure 1(a), there is a grid with 5 rows and 4 columns, and 7 cells are occupied by boxes. After moving every box downwards as much as possible, the boxes are stacked at the bottom of the grid as shown in Figure 1(b).



(a) (b)

Figure 1. A grid with 7 boxes

The moving distance of a box is defined to be the distance the box has traveled until all the boxes are stacked at the bottom of the grid. For example, the moving distance of the highest box at the leftmost column is 2. In this problem, we are interested in the total moving distance of all boxes, that is, the sum of the moving distances of all boxes. In the above example, the total moving distance of the 7 boxes is 8.

You are to calculate the total moving distance of boxes for a given grid.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers, m and n ($1 \leq m, n \leq 100$), where m is the number of rows and n is the number of columns of the grid. In the next m lines, each line contains n integers representing a row of the grid. The information of a grid is given from top row to bottom row. If a cell is occupied by a box, it is represented by the integer `1`, otherwise it is represented by the integer `0`. There is a single space between two integers in the same line.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain the total moving distance of the boxes for the given grid.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 5 4	8 6

1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 3 3 1 1 1 1 1 1 0 0 0 5 6 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1	16

Problem B
Buckets
Time Limit: 2 seconds

You are given two buckets A and B which can hold a and b liters of water, respectively. These buckets do not have any measuring lines and thus you cannot tell exactly how much water is in them. You can only tell if the buckets are full or empty. You are told that the buckets A and B initially hold x and y liters of water, respectively. Assume that there is a reservoir that holds an infinite amount of water. Because the buckets do not have any measuring lines, if you want to keep knowing exactly how much water each bucket holds, all you can do is one of the following operations:

- empty A or B into the reservior,
- fill A or B from the reservior,
- move water from A to B until A is empty,
- move water from A to B until B is full,
- move water from B to A until B is empty, or
- move water from B to A until A is full.

A pair of integers $O_i = (s_i, t_i)$ is called a *target amount*.A target amount is *achievable from* (s, t) if there exists a sequence of zero or more operations from the list above that can make the buckets A and B hold exactly s_i and t_i liters of water respectively, starting from holding s and t liters of water respectively.

We are given a sequence of target amounts $O_1, O_2, O_3, \dots, O_n$ and we are interested in finding the longest subsequence $O_{i_1}, O_{i_2}, O_{i_3}, \dots, O_{i_l}$ ($i_1 < i_2 < \cdots < i_l$ and not necessarily consecutive, i.e., O_1, O_4, O_6, O_8 is allowed if n is large enough), such that each one of them is achievable from the previous one. The previous one for O_{i_1} is defined to be (x, y). In intuitive terms, you are to find the longest subsequence so that you can make the first one from the initial condition using the above operations only, you can make the second one from the first one using the above operations only, and so on.

Given the size of the buckets, the initial amount of water in each bucket, and a sequence of target amounts, write a program to find the longest subsequence of target amouts described above.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing five integers, a, b, x, y, and n ($1 \leq a, b, x, y \leq 1{,}000{,}000{,}000$, $0 \leq x \leq a$, $0 \leq y \leq b$, $1 \leq n \leq 200{,}000$), where a and b are the size of A and B, respectively, x and y are the initial amount of water in A and B, respectively, and n is the number of target amounts given. In the next n lines each, target amounts are given by two integers s and t, which represent the amount of water to be held in A and B, respectively.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain the length of the longest subsequence defined above.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 3 4 1 2 4 1 2 0 3 3 4 3 3 8 2 3 1 7 1 0 0 1 0 0 1 0 0 1 0 2 6 0	4 6

The 38th Annual
ACM International Collegiate
Programming Contest
Asia Regional - Daejeon

Problem C
Geometric Ornaments
Time Limit: 1 second

Mankind has been decorating its objects, its buildings, and itself throughout all of history and back into prehistory. Geometric ornaments are the essential elements of such decoration. We see them every day and everywhere around us. Soohwan, who is a famous computational geometer, has realized that geometric ornaments are very precise geometric constructions and they are usually built on a base of grids and multiple subgrids. Recently, he initiated an ambitious research project on the automatic generation of geometric ornaments.

Among various grids, the rectangular grid and the circular grid have attracted his attention. An $m \times n$ *rectangular grid* is a graph that can be embedded in the plane so that its vertices and edges form a rectangular grid with n vertices appearing in each of m rows and m vertices appearing in each of n columns. More precisely, the $m \times n$ rectangular grid is the graph whose vertex set is $\{v_j^i : 0 \le i \le m-1, 0 \le j \le n-1\}$ and whose edge set is $\{(v_j^i, v_q^p) : |i-p| + |j-q| = 1\}$. The $m \times n$ *circular grid* is obtained from the $m \times n$ rectangular grid by adding so-called wraparound edges (v_{n-1}^i, v_0^i) for every $0 \le i \le m-1$. The 2×6 rectangular grid and the 2×6 circular grid are shown in Figures 1 and 2, respectively.

Some geometric ornaments form spanning trees of the underlying grid, as shown in Figures 1 and 2. Here, a *spanning tree* of a grid refers to a tree, a connected graph without cycles, composed of all the vertices and some edges of the grid. Soohwan wants to enumerate all different *labeled* spanning trees of the $2 \times n$ grid, where the vertices are distinguished from one another by names. (Each vertex is assigned a unique name in a labeled graph to distinguish it from all other vertices.) The two spanning trees of the 2×6 circular grid given in Figure 2 should be counted separately. To support his project, you are to write a program to count the numbers of labeled spanning trees of the $2 \times n$ rectangular grid and of the $2 \times n$ circular grid.

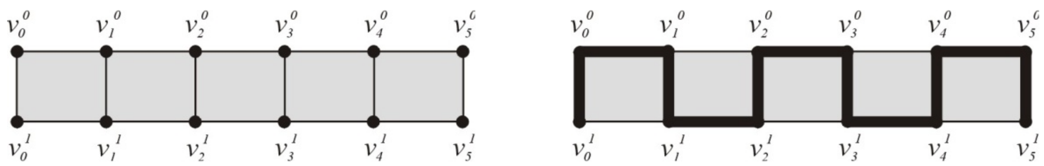


Figure 1. A 2×6 rectangular grid and one of its labeled spanning trees.

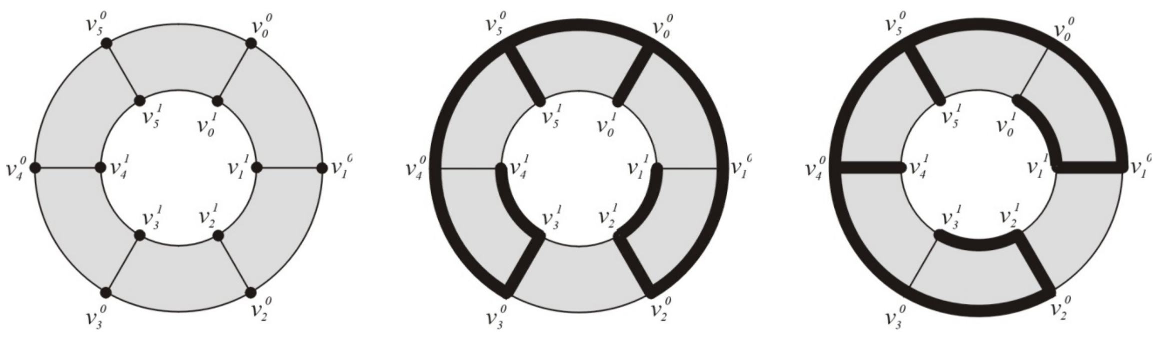


Figure 2. A 2×6 circular grid and its two different labeled spanning trees.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of a single line containing an integer n, indicating that the dimension of the underlying grid is $2 \times n$, where $3 \leq n \leq 50{,}000$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain two integers, R_n modulo $10{,}007$ and C_n modulo $10{,}007$, where R_n and C_n are respectively the numbers of labeled spanning trees of a $2 \times n$ rectangular grid and of a $2 \times n$ circular grid.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 3 4 10	15 75 56 384 1211 9033

Problem D
Golf Field
Time Limit: 2 seconds

Professor Octopus owns a large field which used to be golf course. The field still has many holes into which golf balls were to sink. Professor Octopus offers to sell you a part of his field at a fixed price. The part is determined as follows: you choose four holes in the golf field, and you get the part of the field bounded by the convex hull of the chosen holes. The convex hull of the chosen holes is the smallest convex polygon that contains all the chosen holes in its interior or boundary. We assume that the field is of uniform quality, so you would select four holes such that the area of their convex hull is maximized.

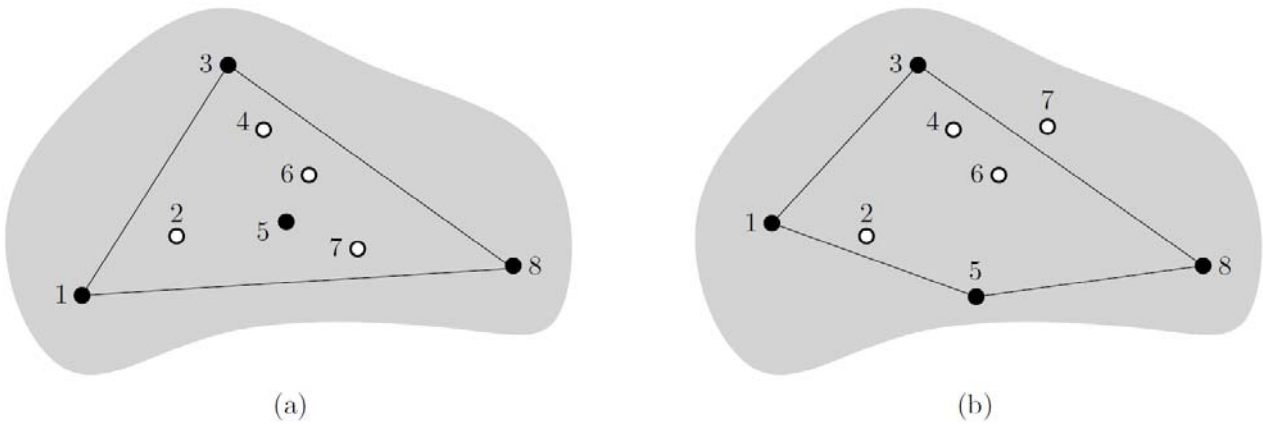


Figure 1

For example, Figure 1 (a) shows eight holes in the field. Imagine that you chose the four holes (black points) 1, 3, 5, and 8. Then any quadrilateral with corners at these four holes is not convex. So the convex hull of these four holes is just the triangle with corners at holes 1, 3, and 8 as shown in the figure. Moreover, it is not difficult to see that this triangle is the largest one among all possible convex hulls of four holes. Figure 1 (b) shows another example of a field with eight holes. In this example, the largest convex hull is the convex quadrilateral with corners at the four holes 1, 3, 5, and 8.

Given a set P of n points in the plane, you write a program to find four points of P whose convex hull has the largest area among all possible convex hulls defined by any four points of P.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with an integer n, the number of points of a set P, where $4 \leq n \leq 30{,}000$. The next line contains a sequence of $2n$ integers, $x_1\ y_1\ x_2\ y_2 \ldots x_n\ y_n$, where x_i and y_i are the x-coordinate and y-coordinate of point p_i of P, respectively. The coordinates are all integers with $-1{,}000{,}000{,}000 \leq x_i \leq 1{,}000{,}000{,}000$ and $-1{,}000{,}000{,}000 \leq y_i \leq 1{,}000{,}000{,}000$. Note that there are no three or more points of P lying on a line.

Output

Your program is to write to standard output. For each test case, print the area of the largest convex hull determined by four points of P. Your output must contain the first digit after the decimal point, rounded off from the second digit.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 4 0 0 3 1 3 3 6 0 6 0 0 2 4 4 1 4 3 6 4 8 0 8 0 4 3 3 4 1 4 7 6 0 6 7 7 5 9 4	9.0 24.0 31.5

Problem E

International Event

Time Limit: 5 seconds

A huge international event is being held every year. There is a long and wide street in front of the event hall. At the beginning of the event, the organizers set up flagpoles beside the street. They raise national flags on the flagpoles and rearrange them every year. This has became the symbol of the event.

The flagpoles are arranged on a line, with flagpole f_i at location ℓ_i. The locations ℓ_i's have integer coordinates and they are all distinct. For the set of nations $\mathbb{N}$, the flag of a nation in $\mathbb{N}$ is raised on a flagpole. In the last year, the flag of nation a_i was raised on flagpole f_i. At the first day of the new year, the nation b_i whose flag is newly raised on the flagpole f_i is determined.

The flag raised on the flagpole f_i should be changed from a_i to b_i. This work is performed by a robot $\mathfrak{R}$ which lowers and raises the flags. The robot $\mathfrak{R}$ can carry an almost infinite number of flags. It may lower the flag of the nation a_i on a flagpole f_i and carry it. Then it may move to the location ℓ_j of flagpole f_j such that $b_j = a_i$ and raise the flag on f_j. This work is always possible because the following condition is satisfied:

For each nation $c \in \mathbb{N}$, the number of flagpoles f_i with $a_i = c$ is equal to the number of flagpoles f_j with $b_j = c$.

There is a special location A different from all ℓ_i's such that the robot $\mathfrak{R}$ should always start and end at A. At the location A, there is no flagpole. Therefore $\mathfrak{R}$ starts at A, delivers all flags of a_i to flagpoles f_j with $b_j = a_i$, and ends at A.

Given the location A and the locations of the flagpoles, and also given the nations a_i and b_i for each flagpole f_i, write a program that computes the minimal travel distance of the robot $\mathfrak{R}$ to deliver all the flags.

For example, in Figure 1, there are six points, representing the flagpoles, and the special point A where the robot $\mathfrak{R}$ starts and ends. The nations of flags correspond to integers in the set $\{1, 2, 3\}$. At each point, a pair of integers (a, b) is given, where a and b are the nations of flags in the last year and in the new year, respectively. The arrows represent a movement of $\mathfrak{R}$ to minimize the travel distance. $\mathfrak{R}$ moves right from A to 5 to load the flag of the nation 2 of the point at 5. While moving left from 5 to 1, it delivers the flags of points at 3 and 5 to the points at 1 and 2, respectively. Then it moves right from 1 to 7 and delivers the flags of points at 1, 2, and 6 to the points at 3, 5, and 7, respectively. Finally, it delivers the flag of the point at 7 to the point at 6 and ends at A, while moving left from 7 to A. The travel distance of $\mathfrak{R}$ is 14.

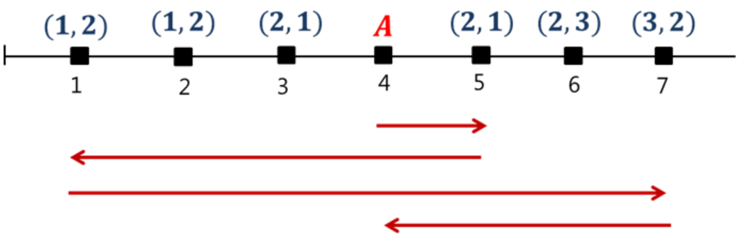


Figure 1.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with an integer N ($2 \leq N \leq 100{,}000$), the number of points representing the flagpoles (not including A). The second line contains an integer α ($1 \leq \alpha \leq 1{,}000{,}000$), the coordinate of A. The third line contains an integer M ($1 \leq M \leq 1{,}000$), representing the set of nations $\{1, 2, \ldots, M\}$. For each integer $i = 1, \ldots, M$, at least one flag of the nation i is raised on a flagpole. The i-th line of the following N lines contains three integers ℓ_i, a_i, and b_i, the coordinate and the nation of the flag of flagpole f_i in the last year and in the new year, respectively. Here $1 \leq \ell_i \leq 1{,}000{,}000$ ($\ell_i \neq \alpha$) and $1 \leq a_i, b_i \leq M$ ($a_i \neq b_i$). Also all ℓ_i's are distinct and given in a nondecreasing order.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the minimum of the travel distance of the robot $\mathfrak{R}$.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 6 4 3 1 1 2 2 1 2 3 2 1 5 2 1 6 2 3 7 3 2 12 12 3 1 2 3 4 1 2 5 2 3 7 1 2 8 2 3 10 1 2 15 2 1 16 2 1 19 3 2 20 2 1 24 3 2 26 3 2	14 62

The 38th Annual
ACM International Collegiate Programming Contest Asia Regional - Daejeon

Problem F
Metal
Time Limit: 5 seconds

We found rare and valuable metals under the ground. The metals are widely scattered underground, so we need a careful plan to dig them. For this, we are going to build a number of horizontal tunnels connected by vertical elevators, as in Figure 1. Every vertical elevator connects the right end of a tunnel to the left end of another tunnel. Furthermore, the connected chain of tunnels and elevators should be monotone to the ground, i.e., when one traverses the chain from the leftmost end of the tunnels to the rightmost end of the tunnels, one passes through all the tunnels without going back to the left.

Due to the limit of the budget, we can construct at most k horizontal tunnels for some positive integer k. Thus at most $k-1$ vertical elevators connecting them are used. We now construct an additional vertical tunnel connecting each metal to the horizontal tunnels. The digging cost of a metal is defined as its vertical distance from the horizontal tunnel. The digging cost for n metals is defined as the maximum digging cost of the metals.

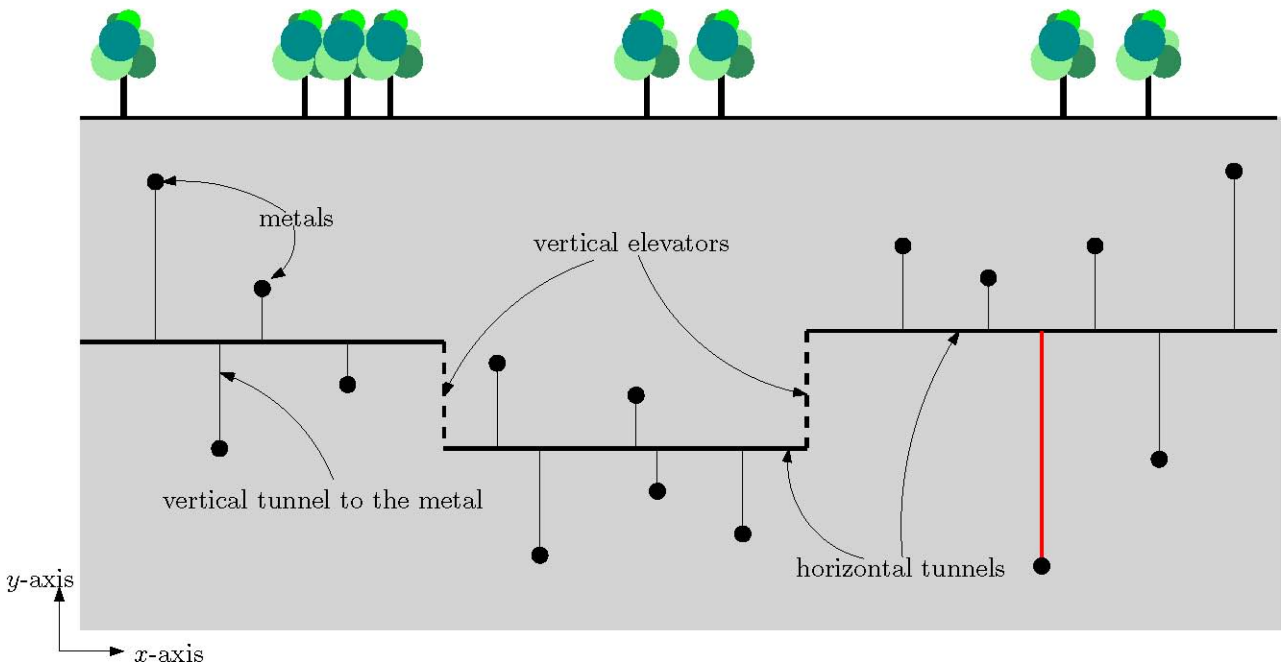


Figure 1. A digging plan for 15 metals for $k = 3$. The vertical distance from the rightmost horizontal tunnel to the lowest metal is the maximum distance in this plan, which determines the digging cost for the metals.

More formally, as in Figure 1, each metal is represented as a point in the two dimensional plane, a horizontal tunnel as a horizontal line segment, a vertical tunnel as a vertical line segment, and an elevator as a vertical line segment. You are given a set of n points, $P = \{p_1, p_2, \cdots, p_n\}$, where p_i has x-coordinate x_i and y-coordinate y_i, and the positive integer k, the maximum number of horizontal tunnels. The digging cost of a metal p_i, $\text{cost}(p_i)$, is the vertical distance to the horizontal tunnels. The digging cost of $P = \{p_1, p_2, \cdots, p_n\}$,

$\text{cost}(P)$, is defined as the maximum cost of $\text{cost}(p_i)$ over all i, i.e., $\text{cost}(P) = \max_{1\le i\le n} \text{cost}(p_i)$. The goal is to determine the position of (at most) k horizontal tunnels such that $\text{cost}(P)$ is minimized.

Given a set P of n points and a positive integer k, you write a program to compute the minimum value of $\text{cost}(P)$.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two positive integers, n ($2 \le n \le 10{,}000$) and k, where n is the number of points (metals) and k is the maximum number of horizontal tunnels to be used. The next line contains $2n$ integers, (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n)$, which respectively represent the coordinates of the points $p_1, p_2, \cdots, p_n$, and $-100{,}000{,}000 \le x_i$, $y_i \le 100{,}000{,}000$ for each i. Two neighboring coordinates are separated by a single space. Note that no two points have the same x-coordinates.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the minimum value of $\text{cost}(P)$. Your output must contain the first digit after the decimal point, rounded off from the second digit.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 3 1 -1 0 0 0 1 0 5 2 2 5 9 2 6 3 -2 2 7 0	0.0 1.5

The 38th Annual

ACM International Collegiate
Programming Contest
Asia Regional - Daejeon

Problem G
Padovan Sequence
Time Limit: 1 second

Consider the spiral of triangles shown in **Figure 1**. It starts with an *equilateral triangle*, that is, a triangle whose side lengths are the same, of side length 1 and it is extended by adding equilateral triangles repeatedly as follows: An equilateral triangle of side length k is added to the longest side of a spiral, where k is the length of the longest side of the spiral.

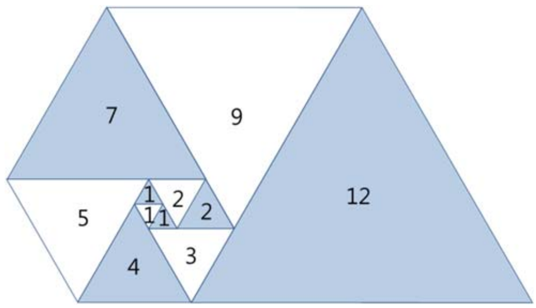


Figure 1. Spiral of equilateral triangles.

Then, the *Padovan sequence* $P(N)$ is the sequence of side lengths of the equilateral triangles in the spiral. The first 10 values $P(1)$ through $P(10)$ are 1, 1, 1, 2, 2, 3, 4, 5, 7, 9.

Given a positive integer N, write a program to compute $P(N)$.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of one line containing an integer N ($1 \leq N \leq 100$).

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain $P(N)$.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 6 12	3 16

The 38th Annual ACM
International Collegiate
Programming Contest
Asia Regional – Daejeon

Problem H
Passwords
Time Limit: 7 seconds

The secret server of ICPC (Inter-Continental Programming Company) uses two passwords, v and w. v and w have the property that $v^{|w|} = w^{|v|}$. That is, if you concatenate v $|w|$ times and w $|v|$ times, they are the same ($|v|$ denotes the length of string v). For example if $v =$ `ab` and $w =$ `abab`, we get $v^4 = w^2 =$ `abababab`. The trivial case where $v = w$ is not allowed, as it is not secure. As it is not easy to memorize the two strings, the system administrator has hidden them into a set of n strings: there are two distinct strings x and y in the set of n strings such that v is a prefix of x ($x = vv'$) and w is a suffix of y ($y = w'w$).

Given the set of strings, write a program which helps finding the password pair.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with an integer n, the number of strings in the set, where $2 \leq n \leq 200$. Each of the following n lines contains one string in English lower case whose length is equal to or smaller than 20,000.

Output
Your program is to write to standard output. Print exactly one line for each test case. Each line should contain two integer values $|v|$ and $|w|$ where $v^{|w|} = w^{|v|}$ for some strings x and y in the set such that v is a prefix of x and w is a suffix of y. Also, $|v| < |w|$. If there are two or more such pairs, print the one where $|v| + |w|$ is the greatest. The uniqueness of such password pair is guaranteed if it exists. If there is no such pair, print `0 0`.

The following shows sample input and output for two test cases.

Sample Input	**Output for the Sample Input**
2 3 abcabe defg bcabab 3 abcdef ghijkl mnopqr	2 4 0 0

Problem I
Permutation Graphs
Time Limit: 5 seconds

A graph G is defined as a tuple of the set of vertices V and the set of edges E, i.e. $G = (V, E)$. The two sets, V and E, are usually given explicitly. For some graphs, however, these sets are given implicitly. A permuation graph is such a graph, where the set of edges is given implicitly.

For instance, consider two permutations (2, 5, 4, 1, 3) and (1, 5, 3, 2, 4) of the numbers $\{1, 2, 3, 4, 5\}$. If we draw line segments connecting the corresponding numbers between two parallel lines, containing the vertices in the two permutation orders, we can find six intersecting pairs of line segments as shown in Figure 1.

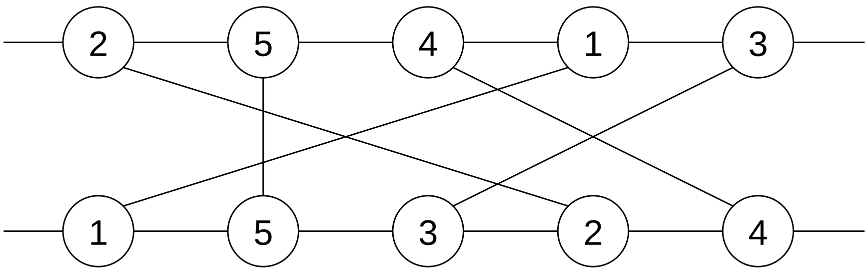


Figure 1: Six intersecting pairs of line segments given two permutations

The intersecting pairs define a permutation graph. Its vertices are the numbers and its edges are the intersecting pairs. In this example, $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 4), (1, 5), (2, 3), (2, 5), (3, 4)\}$. This kind of graphs is called a permutation graph. The two permutations above define the permutation graph as shown in Figure 2.

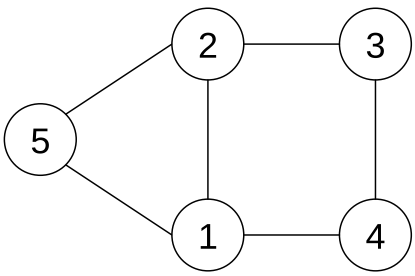


Figure 2: The permutation graph defined by two permutations (2, 5, 4, 1, 3) and (1, 5, 3, 2, 4)

The problem is to find the number of edges of the permutation graph given two permutations of the same set of numbers $\{1, 2, \ldots, n\}$ for some n. For instance, if the two permutations (2, 5, 4, 1, 3) and (1, 5, 3, 2, 4) are given as input, your program should print 6.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. From the second line, each test case is given. A test case consists of three lines. The first line of a test case contains n, the upper bound of the elements of the set $\{1, 2, \ldots, n\}$ for which the permutations are to be defined. The second and the third lines of the test case contain two permutations. The permutations are given as a sequence of natural numbers in $\{1, 2, \ldots, n\}$ separated by one space. The largest possible number n of a permutation is less than or equal to 100,000 ($1 \leq n \leq 100{,}000$).

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the number of edges for the permutation graph corresponding to each test case.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 5 2 5 4 1 3 1 5 3 2 4 7 5 6 7 1 2 3 4 5 6 7 1 2 3 4 7 1 5 3 4 2 7 6 7 1 5 3 4 2 6	6 0 5

Problem J
Set of Rectangles
Time Limit: 2 seconds

A Pythagorean triple consists of three positive integers, a, b, and c, such that $a^2 + b^2 = c^2$. Such a triple is commonly written (a, b, c). A Pythagorean triple (a, b, c) can be generated by two integers x and y ($x > y > 0$) by setting $a = 2xy$, $b = x^2 - y^2$, and $c = x^2 + y^2$.

$\mathbb{R} = \{R_1, R_2, \dots, R_i, \dots\}$ is a set of rectangles. Let w_i and h_i denote the width and height of rectangle R_i, respectively. Also, let d_i denote the length of R_i's diagonal. A rectangle set $\mathbb{R}$ is called a 'Pythagorean Primitive Rectangle Set' if each rectangle R_i in $\mathbb{R}$ holds following constraints:

1. (w_i, h_i, d_i) is a Pythagorean triple;
2. $w_i < h_i$;
3. $\frac{h_i}{w_i} \neq \frac{h_j}{w_j}$ if $i \neq j$.

Bill, a freshman in Pythagoras Memorial High School, got homework in his mathematics class. His homework is described as follows. Given a wire of length L, he should cut it into pieces and bend each piece of the wire to form a rectangle such that the set of rectangles obtained by the cut pieces should be a Pythagorean Primitive Rectangle Set. Note that a piece of length $2(w_i + h_i)$ is required to make rectangle R_i. For example, if rectangle R_i is represented as a pair of its width and its height, (w_i, h_i), and if the total length of the given wire is 94, Bill can cut it into 3 pieces and make a Pythagorean Primitive Rectangle Set, $\mathbb{R} = \{(3,4), (5,12), (8,15)\}$. With the same wire of length 94, Bill can also make another Pythagorean Primitive Rectangle Set, $\mathbb{R} = \{(3,4), (7,24)\}$ in which case there remains a leftover piece. In other words, Bill does not need to use up all the wire to make a Pythagorean Primitive Rectangle Set.

Given a wire of length L, Bill wants to make as many rectangles as possible which are the members of a Pythagorean Primitive Rectangle Set. You are asked to make a program to help Bill.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case has a single integer L ($14 \le L \le 1{,}000{,}000$), the total length of a wire to be cut into pieces.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should show the maximum number of rectangles which can be made by the wire of length L as described above.

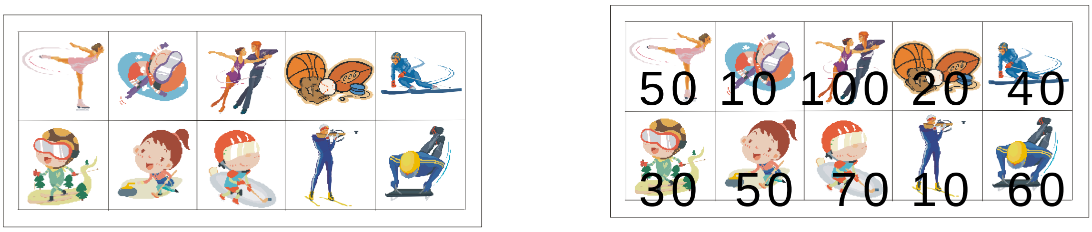
The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 14 1000	1 10

The 38th Annual
ACM International Collegiate
Programming Contest
Asia Regional - Daejeon

Problem K
Stickers
Time Limit: 3 second

Nancy, your little sister, has a sheet of $2n$ stickers of rectangular shape that are arranged in 2 rows and n columns. See Figure 1(a). Nancy wants to decorate her desk with the stickers. But the quality of the stickers is poor, and tearing off one sticker from the sheet spoils the stickers sharing an edge with it. So, Nancy must lose the stickers above, below, to the left of, and to the right of the sticker she tears off.



(a) (b)

Figure 1. A sheet of 10 stickers in 2 rows

Nancy had no idea about what to do. You looked at her and suggested that she should score each sticker and try to choose a possible set of stickers that maximizes the total score. Nancy marked scores to all the $2n$ stickers as in Figure 1(b). And Nancy had no idea, again. You again took a look at her and sighed. You cannot help doing something for her, and at last decided to help her with a fast computer program. Your program is to select a set of stickers of maximum total score from the $2n$ stickers such that no two of them share an edge.

In the example shown in Figure 1, the maximum total score is 260 when you select the four stickers of scores 50, 50, 100, 60. Unfortunately, in this case, it is not allowed to simultaneously select both of the two highest-scored stickers (of score 100 and 70) because the two stickers share an edge between them.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line that contains an integer ($1 \le n \le 100{,}000$), where $2n$ is the number of stickers in the sheet. In the next two lines, each line contains n integers, each of which represents Nancy's score for the sticker at that position in the sticker sheet. Every two consecutive integers in a line are separated by a blank. Note that the $2n$ stickers are of rectangular shape and are arranged in 2 rows and n columns in the sheet. Nancy's scores range from 0 to 100.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain the maximum possible total score for a subset of the $2n$ stickers such that no two stickers share an edge.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 5 50 10 100 20 40 30 50 70 10 60 7 10 30 10 50 100 20 40 20 40 30 50 60 20 80	260 290

Problem L
Term Project
Time Limit: 1 second

Students who enrolled in the 'Problem Solving' course in this fall semester have to carry out a term project. There is no limit to the number of project team members. Even one team is allowed such that all students are members of the same team. In order to form project teams, every student should select a friend with whom he or she wants to work. A student who wants to work alone can select himself or herself. A student list (s_1, $s_2, \ldots, s_r$) can be a team if either $r = 1$ and s_1 selects s_1 or s_1 selects s_2, s_2 selects s_3, … , s_{r-1} selects s_r, and s_r selects s_1.

For example, let's assume that there are 7 students in the class. The students are numbered from 1 to 7. The following is the result of the selection.

1	2	3	4	5	6	7
3	1	3	7	3	4	6

From the above result, we can see that two teams (3) and (4, 7, 6) are formed. Students 1, 2, and 5 don't belong to any team.

Given the result of the selection, write a program to compute the number of students who don't belong to a project team.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer n ($2 \le n \le 100{,}000$), where n is the number of students in the class. All students are numbered from 1 to n. The next line of each test case contains n integers s_1, s_2, … , s_n, where s_i is the student who was a student i selected by.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain the number of students who don't belong to a project team.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 7 3 1 3 7 3 4 6 8 1 2 3 4 5 6 7 8	3 0