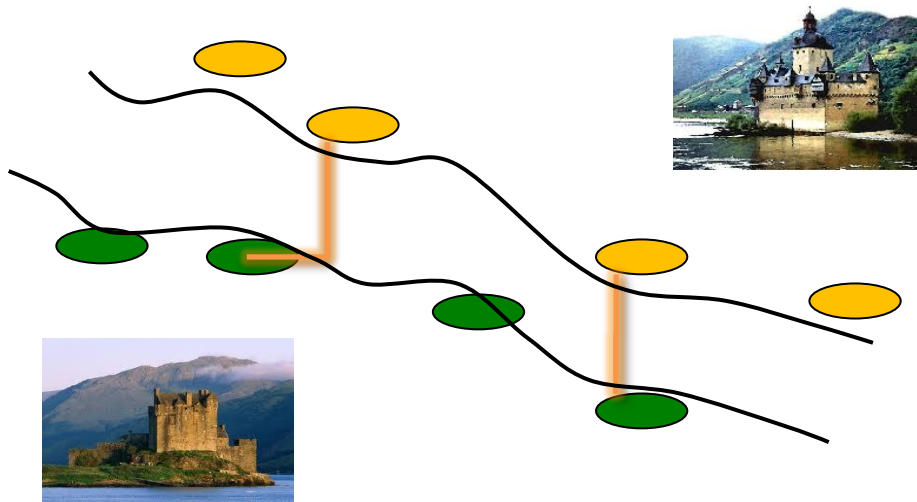


The 36th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition



Problem A Castles

The country ICPCIA has a river called “river of castles”. In the past, ICPCIA was divided into two kingdoms called Westeria and Eastania. Two kingdoms were separated by the river which runs in the direction from northwest to southeast. Both kingdoms had competitively built many castles for defense and attack along the riverside. All castles of each side of the river are located with an x -monotone increasing and y -monotone decreasing feature. In other words, let $S = \{s_1, s_2, \dots, s_n\}$ be the castles of Westeria and $T = \{t_1, t_2, \dots, t_m\}$ the castles of Eastania. Let (x_i, y_i) (resp. (u_i, v_i)) be the coordinates of s_i (resp. t_i). Then $x_i < x_j$ and $y_i > y_j$ if $i < j$. Also, $u_i < u_j$ and $v_i > v_j$ if $i < j$.



Now, the Ministry of Culture and Tourism of ICPCIA decided to build a beautiful bridge connecting two castles on the opposite side of the river. The bridge will be an I-shape (a horizontal segment or a vertical segment) or an L-shape (both of a horizontal segment and a vertical segment). They want to find the location of the bridge such that its length is as small as possible. So, they try to find the closest pair of castles on the opposite side. The distance between two castles s_i and t_j is computed by $|x_i - u_j| + |y_i - v_j|$.

Given the information of two sets of castles, write a program to find the distance between the closest castles on the opposite side of the river.

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 three lines. The first line of each test case contains two integers. The first integer, n , is the number of castles in the west side of the river, and the integer, m , is the number of castles in the east side of it, where $1 \leq n, m \leq 200,000$. The second line of each test case contains $2n$ integers $x_1, y_1, x_2, y_2, \dots, x_n, y_n$, where (x_i, y_i) is the coordinate of the i -th castle in the west side of the river and $x_i < x_j$ and $y_i > y_j$ if $i < j$. The last line of each test case contains $2m$ integers $u_1, v_1, u_2, v_2, \dots, u_m, v_m$, where (u_i, v_i) is the coordinate of the i -th castle in the east side of the river and $u_i < u_j$ and $v_i > v_j$ if $i < j$. You may assume

that there exists an x -monotone increasing and y -monotone decreasing path which separates two sets of castles. All integers representing the coordinates of castles are between -10^9 and 10^9 , inclusive.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the distance between the closest castles on the opposite side of the river.

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

Sample Input	Output for the Sample Input
2	2
4 4	1
1 6 3 4 5 3 6 1	
2 7 6 6 7 4 8 3	
5 4	
1 11 3 8 5 7 6 4 9 3	
3 10 6 8 7 4 10 1	

The 36th Annual ACM
 International Collegiate Programming Contest
 Asia Regional – Daejeon
 Nationwide Internet Competition



Problem B

크로스 컨트리

크로스 컨트리 달리기는 주자들이 자연적인 야외의 지형에 만들어진 코스를 달리는 운동 경기이다. 경주 코스는 일반적으로 4에서 12 킬로미터이며, 숲이나 넓은 땅을 통과하는 풀과 흙으로 된 지면과 언덕과 평평한 지형을 포함한다. 이 경기는 주자들의 개인성적을 매기고, 이를 가지고 팀의 점수를 계산한다.

한 팀은 여섯 명의 선수로 구성되며, 팀 점수는 상위 네 명의 주자의 점수를 합하여 계산한다. 점수는 자격을 갖춘 팀의 주자들에게만 주어지며, 결승점을 통과한 순서대로 점수를 받는다. 이 점수를 더하여 가장 낮은 점수를 얻는 팀이 우승을 하게 된다. 여섯 명의 주자가 참가하지 못한 팀은 점수 계산에서 제외된다. 동점의 경우에는 다섯 번째 주자가 가장 빨리 들어온 팀이 우승하게 된다.

예를 들어, 다음의 표를 살펴보자.

등수	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
팀	A	B	C	C	A	C	B	D	A	A	C	A	C	C	A
점수	1	n/a	2	3	4	5	n/a	n/a	6	7	8	9	10	11	12

팀 B와 D는 선수의 수가 여섯이 아니므로, 점수를 받을 수 없다. 팀 A의 점수는 18 (1+4+6+7)이고, 팀 C의 점수는 18 (2+3+5+8)이다. 이 경우 두 팀의 점수가 같으므로 다섯 번째로 결승점을 통과한 선수를 고려한다. A팀의 다섯 번째 선수의 점수가 C팀의 다섯 번째 선수의 점수보다 적으므로 A팀이 우승팀이 된다.

모든 선수들의 등수가 주어질 때, 우승팀을 구하는 프로그램을 작성하라. 각 팀의 참가 선수가 여섯보다 작으면 그 팀은 점수 계산에서 제외됨을 주의하라. 여섯 명 보다 많은 선수가 참가하는 팀은 없고, 적어도 한 팀은 참가 선수가 여섯이며, 모든 선수는 끝까지 완주를 한다고 가정한다.

입력

입력 데이터는 표준입력을 사용한다. 입력은 T 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 수를 나타내는 정수 T 가 주어진다. 두 번째 줄부터는 두 줄에 하나의 테스트 케이스에 해당하는 데이터가 주어진다. 각 테스트 케이스의 첫 번째 줄에는 하나의 정수 N ($6 \leq N \leq 1,000$)이 주어진다. 두 번째 줄에는 팀 번호를 나타내는 N 개의 정수 t_1, t_2, \dots, t_N 이 공백을 사이에 두고 주어진다. 각 팀은 1과 M ($1 \leq M \leq 200$)사이의 정수로 표현된다.

출력

출력은 표준출력을 사용한다. 하나의 테스트 케이스에 대한 우승팀의 번호를 한 줄에 출력한다.

다음은 두 개의 테스트 데이터에 대한 입력과 출력의 예이다.

Sample Input	Output for the Sample Input
2	1
15	3
1 2 3 3 1 3 2 4 1 1 3 1 3 3 1	
18	
1 2 3 1 2 3 1 2 3 3 3 3 2 2 2 1 1 1	



Problem C Cube of a Graph

The *cube* G^3 of a graph $G = (V, E)$ is the graph on the vertex set V in which two vertices are joined by an edge if their distance in G is at most three, where the distance between two vertices in a graph is defined as the number of edges in a shortest path connecting them. A *bridge* (also known as a cut-edge) of a graph is an edge whose deletion increases the number of connected components. Equivalently, an edge is a bridge if and only if it is not contained in any cycle.

To study hamiltonian properties of the cubes of connected graphs, the following notions concerned with bridge were suggested in early 1960's. A bridge of a graph G is said to be *nontrivial* if neither vertex incident with the edge is of degree one, where the degree of a vertex is the number of edges incident to it. See Figure 1.

- A vertex of G is called a *pure bridge vertex* if each edge incident to the vertex is a nontrivial bridge.
- A set of three distinct mutually adjacent vertices, each of degree at least three, is called a *pure bridge triangle* if each edge of G that is incident with exactly one of three vertices is a nontrivial bridge.

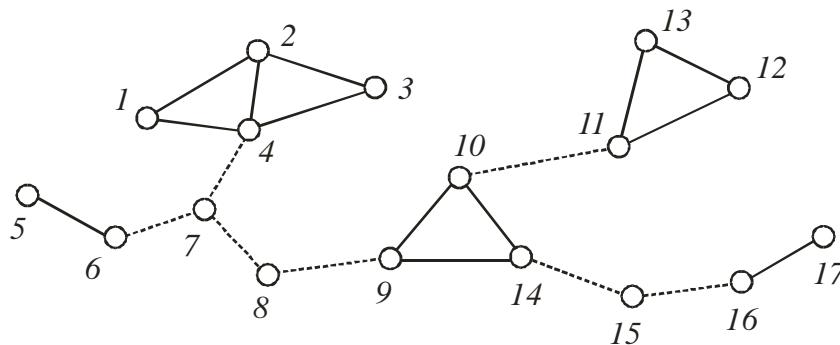


Figure 1. A connected graph which has seven nontrivial bridges represented by dotted lines. There are three pure bridge vertices 7, 8, and 15, one pure bridge triangle $\{9, 10, 14\}$, and one pure bridge pair $\{7, 8\}$.

It has been known that the cube of any connected graph G is hamiltonian-connected, i.e. every two vertices of G^3 are connected by a hamiltonian path. Recently, an ambitious research project on strong hamiltonian properties of graphs was initiated by Prof. Cho, who is a highly considered graph theorist. He introduced a new notion called pure bridge pair to characterize some interesting hamiltonian properties.

- A set of two adjacent vertices is called a *pure bridge pair* if both vertices are pure bridge vertices.

To support Prof. Cho's research, you are to write an efficient running program to identify all the pure bridge vertices, pure bridge triangles, and pure bridge pairs in a large connected graph.

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. The first line of each test case contains two positive integers. The first

integer n is the number of vertices and the second integer m is the number of edges in an input graph G , where $n \leq 3,000$ and $m \leq 1,000,000$. In the following m lines, each line contains two integers u and v which means (u, v) is an edge of G . You may assume that G is connected and the vertex set of G is $\{1, 2, \dots, n\}$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the numbers of pure bridge vertices, pure bridge triangles, and pure bridge pairs in sequence.

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

Sample Input	Output for the Sample Input
2	1 0 0
5 4	3 1 1
2 3	
1 2	
3 4	
5 4	
17 20	
1 2	
2 3	
3 4	
4 1	
2 4	
4 7	
5 6	
6 7	
7 8	
8 9	
9 10	
10 14	
14 9	
10 11	
11 12	
12 13	
13 11	
14 15	
15 16	
16 17	

The 36th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition



Problem D DSLRL

네 개의 명령어 D, S, L, R 을 이용하는 간단한 계산기가 있다. 이 계산기에는 레지스터가 하나 있는데, 이 레지스터에는 0 이상 10,000 미만의 십진수를 저장할 수 있다. 각 명령어는 이 레지스터에 저장된 n 을 다음과 같이 변환한다. n 의 네 자릿수를 d_1, d_2, d_3, d_4 라고 하자(즉 $n = ((d_1 \times 10 + d_2) \times 10 + d_3) \times 10 + d_4$ 라고 하자).

- (1) **D**: D 는 n 을 두 배로 바꾼다. 결과 값이 9999 보다 큰 경우에는 10000 으로 나눈 나머지를 취한다. 그 결과 값($2n \bmod 10000$)을 레지스터에 저장한다.
- (2) **S**: S 는 n 에서 1 을 뺀 결과 $n - 1$ 을 레지스터에 저장한다. n 이 0 이라면 9999 가 대신 레지스터에 저장된다.
- (3) **L**: L 은 n 의 각 자릿수를 왼편으로 회전시켜 그 결과를 레지스터에 저장한다. 이 연산이 끝나면 레지스터에 저장된 네 자릿수는 왼편부터 d_2, d_3, d_4, d_1 이 된다.
- (4) **R**: R 은 n 의 각 자릿수를 오른편으로 회전시켜 그 결과를 레지스터에 저장한다. 이 연산이 끝나면 레지스터에 저장된 네 자릿수는 왼편부터 d_4, d_1, d_2, d_3 이 된다.

위에서 언급한 것처럼, L 과 R 명령어는 십진 자릿수를 가정하고 연산을 수행한다. 예를 들어서 $n = 1234$ 라면 여기에 L 을 적용하면 2341 이 되고 R 을 적용하면 4123 이 된다.

여러분이 작성할 프로그램은 주어진 서로 다른 두 정수 A 와 B ($A \neq B$) 에 대하여 A 를 B 로 바꾸는 최소한의 명령어를 생성하는 프로그램이다. 예를 들어서 $A = 1234, B = 3412$ 라면 다음과 같이 두 개의 명령어를 적용하면 A 를 B 로 변환할 수 있다.

1234 \rightarrow_L 2341 \rightarrow_L 3412
1234 \rightarrow_R 4123 \rightarrow_R 3412

따라서 여러분의 프로그램은 이 경우에 LL 이나 RR 을 출력해야 한다.

n 의 자릿수로 0 이 포함된 경우에 주의해야 한다. 예를 들어서 1000 에 L 을 적용하면 0001 이 되므로 결과는 1 이 된다. 그러나 R 을 적용하면 0100 이 되므로 결과는 100 이 된다.

Input

프로그램은 표준입력으로 입력을 받는다. 프로그램 입력은 T 개의 테스트 케이스로 구성된다. 테스트 케이스 개수 T 는 입력의 첫 줄에 주어진다. 각 테스트 케이스로는 두 개의 정수 A 와 B ($A \neq B$) 가 공백으로 분리되어 차례로 주어지는데, A 는 레지스터의 초기 값을 나타내고 B 는 최종 값을 나타낸다. A 와 B 는 모두 0 이상 10,000 미만이다.

Output

여러분의 프로그램은 표준출력에 출력해야 한다. A 에서 B 로 변환하기 위해 필요한 최소한의 명령어 나열을 출력한다.

다음은 세 개의 테스트케이스에 대한 샘플 입출력이다.

Sample Input	Output for the Sample Input
3 1234 3412 1000 1 1 16	LL L DDDD

The 36th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition



Problem E

Goldbach's Conjecture

A natural number is called a prime number (or a prime) if it is bigger than 1 and has no divisors other than 1 and itself. For example, 5 is prime, since no number except 1 and 5 divides it. On the other hand, 6 is not a prime since $6 = 2 \times 3$.

Goldbach's conjecture is one of the famous unsolved problems in number theory and in all of mathematics. It states: Every even integer greater than 2 can be expressed as the sum of two primes. Such a number is called a Goldbach number. Expressing a given even number as a sum of two primes is called a Goldbach partition of the number. For example, $4 = 2 + 2$, $6 = 3 + 3$, $8 = 3 + 5$, $10 = 7 + 3$ or $10 = 5 + 5$, $12 = 5 + 7$, $14 = 3 + 11$ or $14 = 7 + 7$. Note that Goldbach partition has been found for any even integer n less than 10000.

Given any even integer n greater than 2, write a program that prints the two primes of the Goldbach partition of n . If there are more than one Goldbach partitions of n , find a partition such that the difference of the two primes of it is minimized.

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 an even integer n ($4 \leq n \leq 1,000$).

Output

Your program is to write to standard output. For each test case, find the Goldbach partition as described above, and print its two primes in non-decreasing order with one blank between them.

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

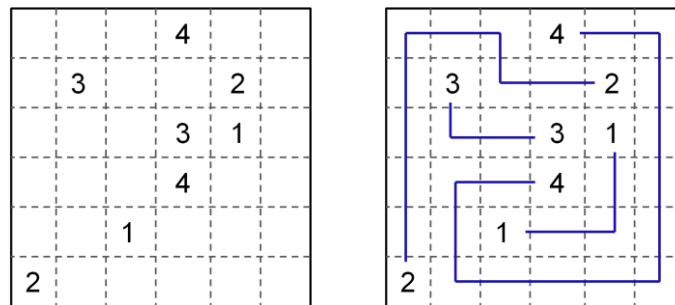
Sample Input	Output for the Sample Input
3	3 5
8	5 5
10	5 11
16	

The 36th Annual ACM
 International Collegiate Programming Contest
 Asia Regional – Daejeon
 Nationwide Internet Competition



Problem F Number Link

Number Link is a type of logic puzzle involving finding paths to connect numbers in a grid. The player has to pair up all the matching numbers on the grid with continuous lines (or paths). The lines cannot branch off or cross over each other, and the numbers have to fall at the end of each line (i.e., not in the middle). Also the lines should pass by each cell in the grid exactly once. An example of Number Link with its solution is shown below.



(a) Initial configuration (b) Solution
 Figure 1. An example of a Number Link Puzzle

Here we concentrate on the case that a pair of only one numbers, say number 1, is located on the grid. Also for the given $m \times n$ grid, m and n are both even. An example for this case is shown in the following Figure 2.

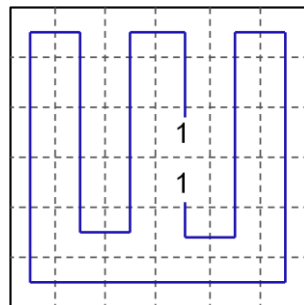


Figure 2. An example in which a pair of number 1's exists

Given an $m \times n$ grid where a pair of number 1's is located, write a program to find a path which connects the number 1's and passes by each cell exactly once.

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. The first line of each test case contains two even integers m and n , the dimensions of the $m \times n$ grid, where $2 \leq m, n \leq 100$. Each of the second and third lines of each test case

contains two integers i and j , representing the location (i, j) of one of the number 1's in the grid. Here the location (α, β) of the number 1 denotes the cell of the α -th row and the β -th column in the grid.

Output

Your program is to write to standard output. For each test case, if there is no path connecting the number 1's, then print -1 . Otherwise, print 1, and in the next mn lines, print the two integers v and w , representing the (v, w) cell on the path. Here the end cells of the path are the locations of the number 1's, that is, print the locations of the number 1's in the first line and the last line, respectively.

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

Sample Input	Output for the Sample Input
2	-1
4 4	1
2 2	4 4
3 3	5 4
6 6	5 5
3 4	4 5
4 4	3 5
	2 5
	1 5
	1 6
	2 6
	3 6
	4 6
	5 6
	6 6
	6 5
	6 4
	6 3
	6 2
	6 1
	5 1
	4 1
	3 1
	2 1
	1 1
	1 2
	2 2
	3 2
	4 2
	5 2
	5 3
	4 3
	3 3
	2 3
	1 3
	1 4
	2 4
	3 4

Problem G Stains

Taeyeon has been using a rectangular carpet for a long time, thus the carpet has lots of indelible stains. Taeyeon decided to repair it by covering the stains with additional carpet patches. The shape of the patches is a diamond, which is a 45°-rotated axis-aligned square, as illustrated in Figure 1 below. The cost of a patch is proportional to its area. Taeyeon wants to minimize the total cost of the patches used to cover the stains, i.e., minimize the sum of area of the patches used. Moreover, Taeyeon is full of artistic sensitivity, so she wants some pattern of the patches. For this, she will align the centers of the patches on a line as illustrated in Figure 1. Patches can overlap each other.

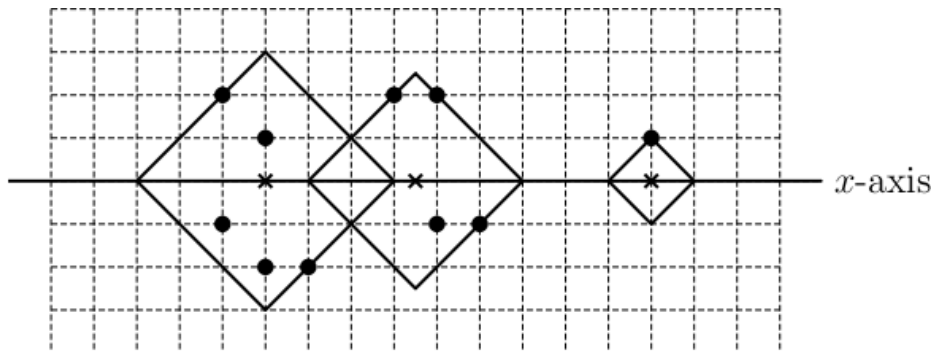


Figure 1. Stains are black dots and the centers of the patches are the crosses. The thick line is the x -axis.

The figure above shows an example in which the nine stains can be covered by three patches with the minimum cost of $32.5 (= 18 + 12.5 + 2)$.

The stains are given with integer coordinates. You should find a set of diamond-shape patches such that (1) every stain must be contained in the interior or on the boundary of some patch, (2) the centers of the patches must be on x -axis, and (3) the sum of area of the patches used must be minimized. You can assume that stains are all distinct and no stains lie on the x -axis.

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 integer n , the number of stains, where $1 \leq n \leq 10,000$. Each of the following n lines contains two integers, representing x -coordinate and y -coordinate of the stains between -10^6 and $+10^6$, inclusive.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a real value, the minimum area of the patches used to cover the stains; you output only 1 digit after decimal point, simply ignore the ones from the 2nd digit after decimal point.

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

Sample Input	Output for the Sample Input
2 10 4 2 14 1 5 -2 5 1 8 2 4 -1 9 2 10 -1 6 -2 9 -1 9 2 1 6 3 11 3 16 6 19 2 20 1 21 3 21 2 24 3	32.5 146.0

The 36th Annual ACM
 International Collegiate Programming Contest
 Asia Regional – Daejeon
 Nationwide Internet Competition



Problem H

연습시즌

두 프로야구팀 X 와 Y가 7 군데의 도시를 정해진 순서로 방문하면서 연습 시즌을 보낸다. 7 개의 도시는 $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ 로 표시되어 있다. 두 팀 X 와 Y 가 연습 시즌을 시작하는 날과 마치는 날은 반드시 동일해야 한다. 이 연습 시즌 동안 각 팀은 도시를 방문해서 연습 훈련을 하거나 또는 한 호텔을 정해서 하루 중일 휴식(휴식일)을 가진다. 각 팀이 방문해야 하는 도시들의 순서는 이미 각각 정해져 있지만 그 중간에 쉬는 휴식일은 따로 정해져 있지 않다. 따라서 방문해야 할 도시들의 순서는 바꿀 수 없지만 그 중간에 휴식일은 상황을 봐가면서 적절히 선택할 수 있다.

예를 들어 연습시즌 동안 한 팀이 방문해야 할 도시의 순서가 $S = \langle c_1, c_2, c_3, c_1, c_4, c_1, c_5, c_4 \rangle$ 와 같다면 각 팀은 그 중간에 휴식일을 넣을 수 있다. 예를 들어 아래 두 개의 일정은 S 에 중간 중간 휴식일이 들어간 상태의 새로운 일정의 예 S1 과 S2 를 보여주고 있다.

$$S1 = \langle c_1, c_2, R, c_3, c_1, R, R, c_4, c_1, c_5, R, c_4 \rangle$$

$$S2 = \langle R, R, c_1, c_2, c_3, c_1, R, c_4, R, c_1, c_5, c_4 \rangle$$

연습시즌에는 많은 비용이 든다. 한 팀이 야구장을 사용할 때 내야 하는 경비는 7 개 도시 모두 동일하게 C 원이다. 만일 두 팀 X 와 Y 가 같은 야구장을 사용해도 전체 비용은 C 이므로 각 팀은 각각 C/2 만큼의 경비만을 내면 되기 때문에 약간 이득이 된다. 즉 두 팀이 다른 야구장에서 따로 훈련을 하면 각각 C 만큼의 경비를 내어 전체 비용은 2C 이지만 같은 야구장을 쓰면 각각 C/2 만 내면 되기 때문에 전체 비용은 C 가 된다. 그런데 각 팀이 호텔에서 휴식일로 쓸 때의 경비계산은 좀 다르다.

연습 시즌 동안 각 팀은 원기회복을 위하여 중간에 휴식일을 가질 수 있다. 이 경우에는 한 호텔을 모두 빌어서 전체가 쓴다. 그런데 호텔에서는 야구팀이 호텔 전체를 빌려서 사용하기 때문에 일단 호텔 전체를 빌릴 경우 독점사용비로 D 만큼 요구한다. 그리고 사용하는 기간에 따라서 추가로 하루당 d 만큼의 비용을 요구한다. 즉 야구팀이 어떤 호텔에 w 일간 연속해서 머물 경우 $D + w * d$ 만큼 지불해야 한다. 예를 들어 전체간 $D=4, d=1$ 인 경우에 한 팀이 5 일간 연속해서 호텔에 머물 경우 전체 비용은 $4 + 1 * 5 = 9$ 만큼의 비용을 내야 한다. 만일 5 일을 각각 따로 따로 떨어진 날로 하루씩 머물 경우에는 모두 5 번의 독점료를 내야 하고 또한 5 일 동안의 사용료를 내야 하므로 전체 비용은 $5 * 4 + 5 * 1 = 25$ 가 된다.

두 팀의 도시 방문순서가 주어졌을 때 특별한 변화 없이 각각 그대로 방문하는 연습시즌 일정을 짜면 아래 표-1 과 같아진다. 이 경우 Y 팀은 앞의 가정대로 연습시즌의 처음과 끝을 X 팀과 맞추기 위해서 마지막 이틀, 7 일과 8 일째 날은 휴식일로 사용하고 있다.

표-1. 간단한 일정-A

	Day1	Day 2	Day 3	Day 4	Day 5	Day 6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	c_3	c_4	c_2	c_6	c_6	c_1	R	R

만일 야구장 사용료가 $C=3$ 이고, 호텔독점료 $D=4$, 일일 사용료 $d=1$ 일 때, 위 표-1 에 있는 일정대로 하면 X 와 Y 가 지불해야 하는 전체 비용은 아래 표-2 와 같다.

표 2. 일정-A 에 따라서 연습시즌을 지낼 경우 비용.

Day	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	c_3	c_4	c_2	c_6	c_6	c_1	R	R
X 비용	3	3	3	3	3	3	3	3
Y 비용	3	3	3	3	3	3	5	1
합	6	6	6	6	6	6	8	4

따라서 X 와 Y 가 지불해야 하는 전체 비용은 $6*6 + 8 + 4 = 48$ 이다. 일정-A 와 조금 다른 다른 일정을 잡아보자. 만일 아래 표-3 과 같이 Y 팀의 일정에 휴식일을 적절히 넣어서 조정하면 두 팀이 야구장을 공동으로 쓰는 날이 늘어나므로 전체 비용을 줄일 수 있다.

표 3. 일정-B

Day	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8
X	c_1	c_3	c_4	c_5	c_2	c_6	c_6	c_1
Y	R	c_3	c_4	R	c_2	c_6	c_6	c_1
X 비용	3	1.5	1.5	3	1.5	1.5	1.5	1.5
Y 비용	5	1.5	1.5	5	1.5	1.5	1.5	1.5
Total	8	3	3	8	3	3	3	3

따라서 전체 비용은 $3*6 + 8*2 = 34$ 이 되어 일정-A 에 따른 비용에 비해서 $48-34 = 14$ 만큼 절약할 수 있다. 그러나 최적의 일정은 C, D 그리고 d 값에 따라서 다르게 결정될 수 있음을 유의해야 한다.

여러분은 X 와 Y 팀의 방문도시 일정이 주어졌을 때 두 팀이 지출해야 하는 비용의 합을 최소로 가는 가장 좋은 일정을 구하고 그 전체 비용을 출력하는 것이다. 아래 입출력의 예를 잘 살펴보면 이 문제를 잘 이해할 수 있을 것이다.

Input

입력은 표준 입력으로 들어온다. 입력은 모두 T 개의 테스트 케이스가 있다. 테스트 케이스의 전체 개수는 첫 줄에 주어진다. 각 테스트 케이스는 3 개의 줄로 이루어져 있다. 첫 라인에는 C, D, d 를 나타내는 정수 값이 차례대로 하나의 공백을 두고 나타난다. 각 숫자는 모두 10 이하의 양의 정수이다. 그 다음 이어지는 두 개의 줄에는 방문해야 할 도시의 순서가 하나의 공백을 두고 나타난다. 각 도시는 1 부터 7 까지의 정수로 표시된다. 그리고 그 끝은 숫자 0 으로 표시되어 있다. 방문해야 할 도시의 개수 N 은 $2 < N < 100$ 로 제한되어 있다.

Output

여러분의 프로그램은 표준출력에 써야 한다. 각 테스트 케이스에 해당되는 정답, 즉 연습시즌을 마칠 수 있는 최소의 비용(X 와 Y 비용의 합)을 출력해야 한다.

다음은 다섯 개의 테스트 데이터에 대한 입력과 출력의 예이다.

Sample Input	Output for the Sample Input
5	27
4 5 2	92
3 5 6 7 5 0	28
3 5 6 7 0	115
10 5 1	51
1 2 3 4 5 6 7 0	
2 3 4 5 6 7 1 0	
2 9 1	
1 2 3 4 5 6 7 0	

2 3 4 5 6 7 1 0	
8 1 3	
1 2 3 4 5 6 7 6 5 4 3 0	
2 5 4 0	
4 3 3	
5 6 7 1 4 5 6 7 0	
3 5 6 7 0	

The 36th Annual ACM
 International Collegiate Programming Contest
 Asia Regional – Daejeon
 Nationwide Internet Competition



Problem I 두 수의 합

여러 개의 서로 다른 정수 $S = \{a_1, a_2, \dots, a_n\}$ 와 또 다른 정수 K 가 주어졌을 때, S 에 속하는 서로 다른 두 개의 정수의 합이 K 에 가장 가까운 두 정수를 구하시오. 예를 들어, 10 개의 정수

$$S = \{-7, 9, 2, -4, 12, 1, 5, -3, -2, 0\}$$

가 주어졌을 때, $K = 8$ 에 그 합이 가장 가까운 두 정수는 $\{12, -4\}$ 이다. 또한 $K = 4$ 에 그 합이 가장 가까운 두 정수는 $\{-7, 12\}, \{9, -4\}, \{5, -2\}, \{5, 0\}, \{1, 2\}$ 등의 다섯 종류가 있다.

여러 개의 서로 다른 정수가 주어졌을 때, 주어진 정수들 중에서 서로 다른 두 정수의 합이 주어진 또 다른 정수에 가장 가까운 두 정수의 조합의 수를 계산하는 프로그램을 작성하시오.

입력

프로그램은 표준입력으로 입력을 받는다. 프로그램 입력은 t 개의 테스트 케이스로 구성된다. 입력의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 두 줄에 한 개의 테스트 케이스에 해당하는 데이터가 주어진다. 각 테스트 케이스의 첫 번째 줄에는 두 개의 정수 n 과 K ($2 \leq n \leq 1,000,000$, $-10^8 \leq K \leq 10^8$)가 한 개의 공백을 사이에 두고 입력된다. 두 번째 줄에는 n 개의 정수가 하나의 공백을 사이에 두고 주어지며, 각 정수의 최대값은 10^8 이고, 최소값은 -10^8 이다. 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스의 출력되는 첫 줄에 입력으로 주어진 n 개의 정수들 중에서 서로 다른 두 정수의 합이 주어진 또 다른 정수 K 에 가장 가까운 두 정수의 조합의 수를 출력한다.

다음은 네 개의 테스트 데이터에 대한 입력과 출력의 예이다.

Sample Input	Output for the Sample Input
4	1
10 8	5
-7 9 2 -4 12 1 5 -3 -2 0	1
10 4	2
-7 9 2 -4 12 1 5 -3 -2 0	
4 20	
1 7 3 5	
5 10	
3 9 7 1 5	

The 36th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition



Problem J Widest Path

We are given a graph which represents connections between nodes in the computer network, and the weight of an edge represents the bandwidth of a connection between two nodes. For the efficient data transmission between two nodes in the network, we are interested in finding a path between two nodes that has wide bandwidth. The *bandwidth of a path* is the minimum weight of an edge in the path. The widest path problem is to find the path between two nodes that has the maximum possible bandwidth.

For example, the widest path from node 1 to node 4 in Figure 1 has bandwidth 25, and passes through node 3 and node 2. The widest path from node 6 to node 3 has bandwidth 30, and passes through node 5.

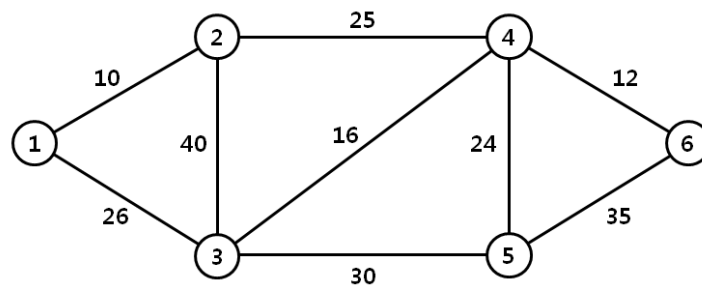


Figure 1. Example of a computer network

Given two nodes in a graph, write a program which determines the bandwidth of the widest path between two nodes.

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 four integers n , m , s and t for a connected graph, where n ($2 \leq n \leq 1,000$) represents the number of nodes and m ($1 \leq m \leq \frac{n(n-1)}{2}$) represents the number of edges, s and t ($s \neq t$) represents the two nodes (nodes are numbered from 1 to n). In the following m lines, the bandwidth of the edges are given; each line contains three integers, u , v , and b , where b ($1 \leq b \leq 10^5$) is the bandwidth of a connection between two nodes u and v .

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the bandwidth of the widest path between two nodes s and t .

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

Sample Input

```
2
6 9 1 4
1 2 10
3 1 26
2 3 40
2 4 25
4 3 16
3 5 30
4 5 24
4 6 12
6 5 35
5 4 5 2
1 2 4
2 3 3
3 4 2
4 5 1
```

Output for the Sample Input

```
25
1
```