The 39th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Korea Nationwide Internet Competition

Problem A

ACM Hotel

Time Limit: 1 Second

Jiwoo, the manager of the ACM Hotel, is about to assign the vacant rooms to the guests upon their arrival. According to customers' survey, the customers prefer the rooms which are close to the main entrance on-walk. Jiwoo likes to assign the rooms on this policy. Write a program to help Jiwoo on assigning the rooms for the guests.

For simplicity, let's assume that the ACM hotel is a rectangular shape, an H story building with W rooms on each floor ($1 \leq H, W \leq 99$) and that the only one elevator is on the leftmost side (see Figure 1). Let's call this kind of hotel as $H \times W$ shaped. The main entrance is located on the first floor near the elevator. You may ignore the distance between the gate and the elevator. Also assume that the distances between neighboring rooms are all the same, the unit distance, and that all the rooms only in the front side of the hotel.

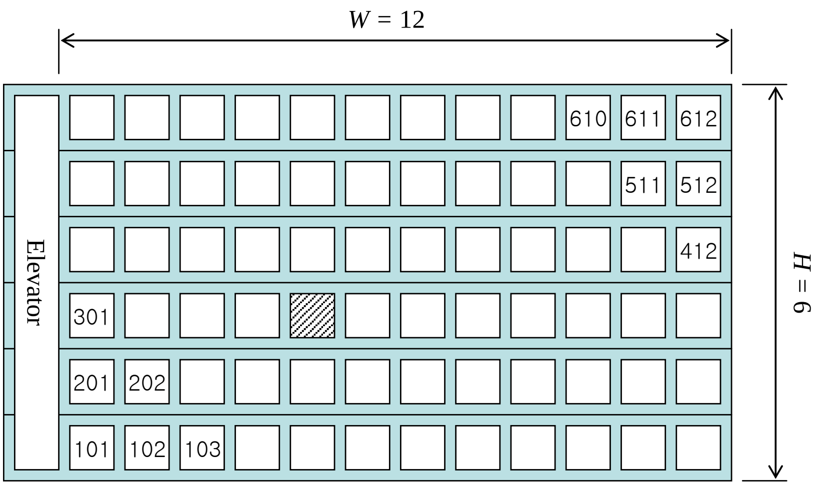


Figure 1. An abstract view of an $\boldsymbol{H} \times \boldsymbol{W}$ hotel where $\boldsymbol{H}$ = 6 and $\boldsymbol{W}$ = 12

The rooms are numbered in *YXX* or *YYXX* style where *Y* or *YY* denotes the number of the floor and *XX*, the index of the room counted from the left. Therefore the room shaded in Figure 1 should be 305.

The customers do not concern the distance moved in the elevator though the room on the lower floor is preferred than that on the higher floor if the walking distance is same. For instance, the room 301 is preferred than the room 102 since the customer should walk for two units for the latter but one unit, for the former. Additionally, the room 2101 is preferred than the room 102.

Your program should compute the room number which should be assigned for the N-th guest according to this policy assuming that all the rooms are vacant initially. The first guest should be assigned to 101, the second guest to 201, and so on. In Figure 1, for example, the 10th guest should be assigned to the room 402 since $H = 6$.

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. Each test case consists of a single line containing and integers H, W, and N: the number of floors, the number of rooms on each floor, and the index of the arrival time of the guest to be assigned a room, respectively, where $1 \leq H, W \leq 99$ and $1 \leq N \leq H \times W$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the room number of the given hotel, where the N-th guest should be assigned.

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

Sample Input	Output for the Sample Input
2 6 12 10 30 50 72	402 1203

Problem A

ACM 호텔

Time Limit: 1 Second

ACM 호텔 매니저 지우는 손님이 도착하는 대로 빈 방을 배정하고 있다. 고객 설문조사에 따르면 손님들은 호텔 정문으로부터 걸어서 가장 짧은 거리에 있는 방을 선호한다고 한다. 여러분은 지우를 도와 줄 프로그램을 작성하고자 한다. 즉 설문조사 결과 대로 호텔 정문으로부터 걷는 거리가 가장 짧도록 방을 배정하는 프로그램을 작성하고자 한다.

문제를 단순화하기 위해서 호텔은 직사각형 모양이라고 가정하자. 각 층에 W 개의 방이 있는 H층 건물이라고 가정하자 ($1 \le H, W \le 99$). 그리고 엘리베이터는 가장 왼쪽에 있다고 가정하자(그림 1 참고). 이런 형태의 호텔을 $H \times W$ 형태 호텔이라고 부른다. 호텔 정문은 일층 엘리베이터 바로 앞에 있는데, 정문에서 엘리베이터까지의 거리는 무시한다. 또 모든 방 사이의 거리는 같은 거리(거리 1)라고 가정하고 호텔의 정면 쪽에만 방이 있다고 가정한다.

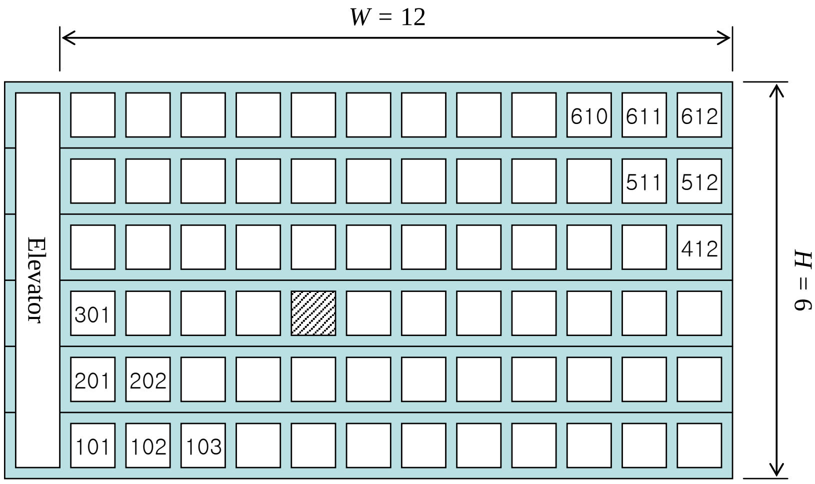


그림 1. $H = 6$ 이고 $W = 12$ 인 $H \times W$ 호텔을 간략하게 나타낸 그림

방 번호는 YXX 나 $YYXX$ 형태인데 여기서 Y 나 YY 는 층 수를 나타내고 XX 는 엘리베이터에서부터 세었을 때의 번호를 나타낸다. 즉, 그림 1 에서 빗금으로 표시한 방은 305 호가 된다.

손님은 엘리베이터를 타고 이동하는 거리는 신경 쓰지 않는다. 다만 걷는 거리가 같을 때에는 아래층의 방을 더 선호한다. 예를 들면 102 호 방보다는 301 호 방을 더 선호하는데, 102 호는 거리 2 만큼 걸어야 하지만 301 호는 거리 1 만큼만 걸으면 되기 때문이다. 같은 이유로 102 호보다 2101 호를 더 선호한다.

여러분이 작성할 프로그램은 초기에 모든 방이 비어있다고 가정하에 이 정책에 따라 N 번째로 도착한 손님에게 배정될 방 번호를 계산하는 프로그램이다. 첫 번째 손님은 101 호, 두 번째 손님은 201 호 등과 같이 배정한다. 그림 1 의 경우를 예로 들면, $H = 6$이므로 10 번째 손님은 402 호에 배정해야 한다.

Input

프로그램은 표준 입력에서 입력 데이터를 받는다. 프로그램의 입력은 T 개의 테스트 데이터로 이루어져 있는데 T 는 입력의 맨 첫 줄에 주어진다. 각 테스트 데이터는 한 행으로서 $\boldsymbol{H}, \boldsymbol{W}, \boldsymbol{N}$, 세 정수를 포함하고 있으며 각각 호텔의 층 수, 각 층의 방 수, 몇 번째 손님인지를 나타낸다($\mathbf{1} \leq \boldsymbol{H}, \boldsymbol{W} \leq \mathbf{99}$, $\mathbf{1} \leq \boldsymbol{N} \leq \boldsymbol{H} \times \boldsymbol{W}$).

Output

프로그램은 표준 출력에 출력한다. 각 테스트 데이터마다 정확히 한 행을 출력하는데, 내용은 N 번째 손님에게 배정되어야 하는 방 번호를 출력한다.

두 개의 테스트 데이터에 대한 입출력 예는 다음과 같다.

Sample Input	Output for the Sample Input
2 6 12 10 30 50 72	402 1203

The 39th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Korea Nationwide Internet Competition

Problem B
Driving License
Time Limit: 1 Second

Taking a driving test must make people nervous in almost every city in the world. People in Grid city in which road network looks like a grid should pass an interesting driving test in order to obtain a driver's license. Information on an $M \times N$ grid-shaped road network is given to the applicant before he takes the test in the city. Using the given information, he has to comply with 3 rules, described below, to pass the test.

Rule1: He must drive a car eastward or downward from the starting point s, located at upper-left corner of the grid, to the ending point t, located at lower-right corner of the grid.

Rule 2: The driving time for each segment (either vertical or horizontal) of the grid must be exactly L (which is given before taking the test) and the turning time to change direction must be exactly 1. No turning time is assumed at the starting point s.

Rule 3: Using an amount of fuel, say G, given before taking the test, he has to reach t as soon as possible without consuming additional fuel.

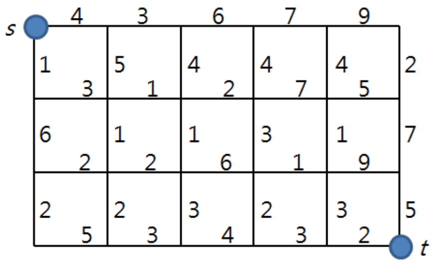


Figure 1.

Note that he has to control the speed of the car considering the road condition to comply with the rule 2. So, the amount of fuel consumed to pass each segment could be different. Also note that he has to find a fastest route among several possible routes along which he can drive using as much fuel as G.

Fig. 1 shows a 4×6 grid-shaped road network, which has 4 horizontal streets and 6 vertical streets. Numbers shown for each segment of the grid denote the required amount of fuel to pass it.

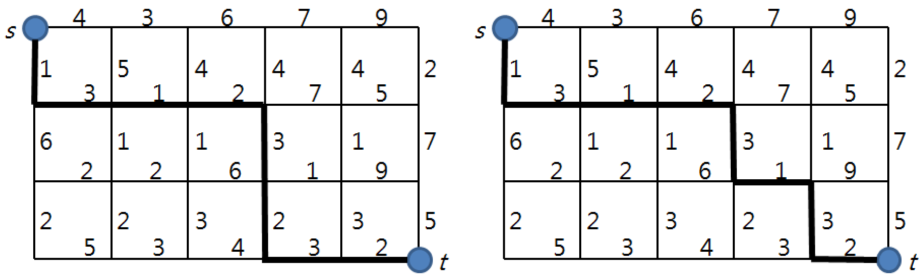


Figure 2.

Suppose a road network as shown in Fig. 1 and $G = 19$ and $L = 10$ are given as input. Fig. 2 shows two possible routes. The route shown in the left figure consumes fuel as much as 17 and it takes driving time 83 since it passes 8 grid segments and changes direction 3 times. On the other hand, the route shown in the right figure consumes a bit less fuel but it takes driving time 85. Note that there are some other routes along which he can drive with fuel 19 within time 83. However, there is no route along which he can drive faster with fuel 19.

You are to make a program which helps to find a route to pass the driving test in Grid city. You can assume that given a grid, the starting point locates at upper-left corner and the ending point at lower-right corner and that no fuel is consumed for turning direction.

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 4 integers: $\boldsymbol{M}$, $\boldsymbol{N}$, $\boldsymbol{L}$, and $\boldsymbol{G}$, where $\boldsymbol{M}$ and $\boldsymbol{N}(\mathbf{2} \le \boldsymbol{M}, \boldsymbol{N} \le \mathbf{100})$ denote the dimension of the grid, $\boldsymbol{L}(\mathbf{1} \le \boldsymbol{L} \le \mathbf{10})$ the driving time to pass each grid segment, and $\boldsymbol{G}(\mathbf{1} \le \boldsymbol{G} \le \mathbf{1,000,000})$ the amount of fuel given before testing. Each of the following $\boldsymbol{M}$ lines contains $\boldsymbol{N} - \mathbf{1}$ integers, each of which denotes the amount of fuel consumed to pass the corresponding horizontal segment of the grid. Another $\boldsymbol{M} - \mathbf{1}$ lines follow. Each of the following $\boldsymbol{M} - \mathbf{1}$ lines contains $\boldsymbol{N}$ integers, each of which denotes the amount of fuel consumed to pass the corresponding vertical segment of the grid. Each integer used for the amount of fuel to pass each correspoing segment of the grid is within 1 and 1,000 inclusive.

All the integers shown in a same line are separated by a blank.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should show the total elapsed time along the fastest route. If it is impossible to drive from s to t along any route with given amount of fuel G, print -1.

The following shows sample input and output for three test cases.
(Note: the first test case corresponds to Figure 1)

Sample Input	Output for the Sample Input
3 4 6 10 19 4 3 6 7 9 3 1 2 7 5 2 2 6 1 9 5 3 4 3 2 1 5 4 4 4 2 6 1 1 3 1 7 2 2 3 2 3 5 3 4 5 10 4 5 6 2 3 1 5 7 8 1 8 6 7 4 6 9 1 3 3 10 9 2 2 2 2 2 2 3 3 3 3 3 3	83 27 -1

The 39th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Korea Nationwide Internet Competition

Problem C
Grid Graphs
Time Limit: 1 Second

The $m \times n$ *rectangular grid* is a graph whose vertices correspond to the points in the plane with x-coordinates being integers in the range $0, \dots, n-1$ and y-coordinates being integers in the range $0, \dots, m-1$, and two vertices are joined by an edge whenever the corresponding points are at unit distance apart. For example, a 4×6 rectangular grid is shown in Figure 1. The grid has n vertices appearing in each of m rows and m vertices in each of n columns. The vertex in row i and column j is denoted by (i, j), where $0 \le i \le m-1$ and $0 \le j \le n-1$.

If we add an edge joining two vertices $(i, 0)$ and $(i, n-1)$ of the $m \times n$ rectangular grid for every row $i \in \{0, \dots, m-1\}$ and moreover, add an edge between two vertices $(0, j)$ and $(m-1, j)$ for every column $j \in \{0, \dots, n-1\}$, then each row forms a cycle of length n and each column forms a cycle of length m, as illustrated in Figure 2. The resulting graph is often called an $m \times n$ *toroidal grid*, because it can be drawn on a torus without edge crossings.

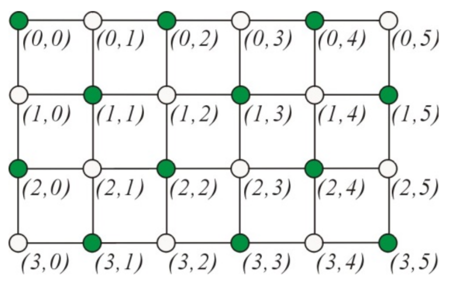


Figure 1. A 4×6 rectangular grid.

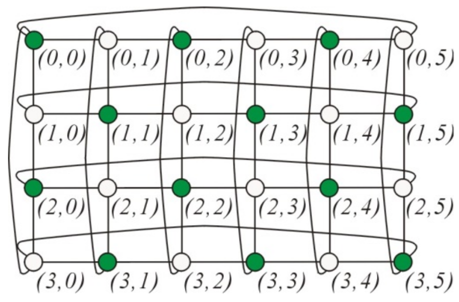


Figure 2. A 4×6 toroidal grid.

Given an $m \times n$ toroidal grid, you are to write a program to find a cycle that visits every vertex exactly once. Here, the required cycle may be represented as a sequence, $(v_1, v_2, \dots, v_{mn})$, of mn distinct vertices of the graph such that v_k and v_{k+1} are adjacent for all $k \in \{1, \dots, mn-1\}$ and moreover, v_{mn} and v_1 are adjacent.

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 two integers, m and n, where $3 \le m, n \le 100$, indicating that the input graph is an $m \times n$ toroidal grid.

Output

Your program is to write to standard output. For each test case, the first line of the results must contain an integer indicating whether there exists a feasible solution. If yes, the integer must be `1`; otherwise `-1`. If the first line is `1` and only `1`, then it must be followed by mn lines, describing a sequence of vertices of the required cycle. In case multiple solutions are possible, just output any one of them. No whitespace characters (blanks and/or tabs) are allowed inside a line.

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

Sample Input	Output for the Sample Input
2 3 4 3 3	1 (0,0) (0,1) (1,1) (1,2) (0,2) (0,3) (1,3) (2,3) (2,2) (2,1) (2,0) (1,0) 1 (2,2) (2,0) (1,0) (0,0) (0,1) (0,2) (1,2) (1,1) (2,1)

Problem C
그리드 그래프
Time Limit: 1 Second

$m \times n$ *직사각 그리드(rectangular grid)*는, x-좌표의 범위가 0부터 $n-1$까지인 정수이고 y-좌표의 범위가 0부터 $m-1$까지 정수인 평면상의 점들에 대응하는 정점들을 가지고, 두 정점에 대응하는 두 점 사이의 거리가 1 일 때에만 그 둘을 잇는 에지가 있는 그래프다. 예를 들어, 4×6 직사각 그리드가 그림 1 에 있다. 이 그리드는 m개 행 각각에 n개의 정점이 있고, n개 열 각각에 m 개의 정점이 있다. 행 i와 열 j에 있는 정점을 (i,j)라고 나타낸다. 여기서 $0 \le i \le m-1$이고 $0 \le j \le n-1$이다.

$m \times n$ 직사각 그리드의 모든 행 $i \in \{0,\dots,m-1\}$에 대하여 두 정점 $(i,0)$과 $(i,n-1)$을 잇는 에지를 추가하고, 또한 모든 열 $j \in \{0,\dots,n-1\}$에 대하여 두 정점 $(0,j)$와 $(m-1,j)$을 잇는 에지를 추가하면, 그림 2 에 보인 것과 같이 각 행은 길이 n인 사이클을 이루고 각 열은 길이 m인 사이클을 이루게 된다. 이렇게 만들어진 그래프를 종종 $m \times n$ *토로이드 그리드(toroidal grid)* 라고 부르는데, 왜냐하면 이 그래프를 토러스(torus)에 에지가 교차하지 않도록 그릴 수 있기 때문이다.

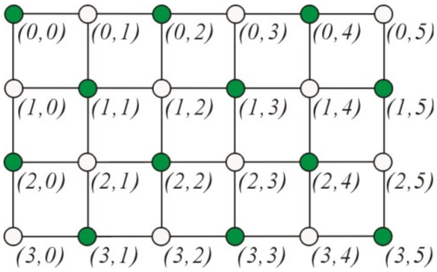


그림 1. 4×6 직사각 그리드.

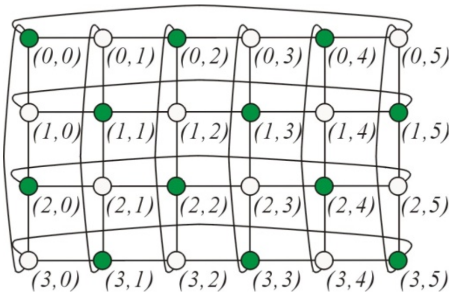


그림 2. 4×6 토로이드 그리드

주어진 $m \times n$ 토로이드 그리드에 대하여, 모든 정점을 정확히 한번씩 지나는 사이클을 찾는 프로그램을 작성하시오. 문제에서 요구하는 사이클은 그래프에 있는 서로 다른 mn개 정점들의 열 $(v_1, v_2, \dots, v_{mn})$로 나타낼 수 있는데, 이때 모든 $k \in \{1,\dots,mn-1\}$에 대하여 v_k와 v_{k+1}은 인접하며 또한 v_{mn}과 v_1도 인접하여야 한다.

입력(Input)

표준입력에서 데이터를 읽는다. 입력은 T개의 테스트 데이터로 구성되어 있다. 입력의 첫째 줄에 입력 데이터의 개수를 나타내는 정수 T가 주어진다. 각 테스트 데이터는 두 정수 m과 n을 포함하는 한 줄로 구성되어 있는데, 입력 그래프가 $m \times n$ 토로이드 그리드임을 가리킨다. 여기서 $3 \le m, n \le 100$이다.

출력(Output)

표준출력으로 데이터를 출력한다. 각 테스트 데이터에 대해, 출력의 첫째 줄에는 조건을 만족하는 해가 존재하는지 아닌지를 나타내는 정수를 출력해야 한다. 만약 해가 존재하면, 그 정수는 1 이어야 한다; 그렇지 않으면 -1 이다. 첫째 줄이 1 일 경우에만 추가로 문제에서 요구한 사이클의 정점 열(sequence)을 mn 개의 줄에 출력한다. 복수의 해가 가능하면, 그 중 임의의 하나를 출력하면 된다. 어떤 줄에도 공백 문자(빈칸이나 탭)는 허용되지 않는다.

다음은 두 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력 예제(Sample Input)	**출력 예제(Output for the Sample Input)**
2 3 4 3 3	1 (0,0) (0,1) (1,1) (1,2) (0,2) (0,3) (1,3) (2,3) (2,2) (2,1) (2,0) (1,0) 1 (2,2) (2,0) (1,0) (0,0) (0,1) (0,2) (1,2) (1,1) (2,1)

Problem D
Henry
Time Limit: 1 Second

Henry, a 10 year old boy, has just learned about the concept of fractions today from his math teacher Ahmes. Henry liked so much to play with fractions, and all of a sudden found it funny to represent a fraction by the sum of different fractions. Henry has tried many fractions, and discovered that all fractions he tried can be represented by a sum of *distinct* unit fractions. A unit fraction is a fraction whose numerator is 1. For example, $\frac{4}{23}$ can be represented as $\frac{1}{6}+\frac{1}{138}$.

Henry told his discovery to Ahmes. Ahmes was so happy about the little boy's brilliant discovery, and named such a representation of a fraction a *Henry representation*. That is, a Henry representation of fraction $\frac{a}{b}$ is a series of distinct unit fractions that sum up exactly to $\frac{a}{b}$. Henry and Ahmes continued studying Henry representations. At last they found that every fraction that is smaller than 1 indeed has a Henry representation and may have many different ones: for example, $\frac{5}{7}=\frac{1}{2}+\frac{1}{5}+\frac{1}{70}=\frac{1}{2}+\frac{1}{6}+\frac{1}{21}=\frac{1}{2}+\frac{1}{7}+\frac{1}{14}$. Also, notice by definition that $\frac{2}{3}=\frac{1}{3}+\frac{1}{3}$ is not a Henry representation because that contains two equal unit fractions $\frac{1}{3}$.

Ahmes and Henry also found an easy procedure to find a Henry representation of a given fraction, described as follows: given a fraction $\frac{a}{b}$ where a and b are positive integers with $a<b$, choose the largest unit fraction $\frac{1}{x_1}$ such that $\frac{1}{x_1}\le\frac{a}{b}$. Then, subtract $\frac{1}{x_1}$ from $\frac{a}{b}$, and repeat the process for the remainder: choose the largest unit fraction $\frac{1}{x_2}$ such that $\frac{1}{x_2}\le\frac{a}{b}-\frac{1}{x_1}$. Repeat again until there is no remainder. This procedure will result in a series of unit fractions $\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\ldots$ that sum up to the given fraction $\frac{a}{b}$. Ahmes and Henry proved that this procedure always terminates for any input fraction $\frac{a}{b}$ with a finite number of distinct unit fractions, that is, a Henry representation.

Ahmes and Henry want to have their algorithm implemented by you. Given a fraction $\frac{a}{b}$, you are to write a computer program that prints out the denominator of the last fraction in the resulting Henry representation of their algorithm. For example, if $\frac{a}{b}=\frac{5}{7}$, then your program must print out 70 since the algorithm will terminate with a Henry representation $\frac{5}{7}=\frac{1}{2}+\frac{1}{5}+\frac{1}{70}$.

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 line containing two integers a and b ($1\le a<b\le 10{,}000$) that are relatively prime, representing the input fraction $\frac{a}{b}$. Note that if $\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\ldots,\frac{1}{x_m}$ are the unit fractions produced by Ahmes and Henry's algorithm in order for the input $\frac{a}{b}$, then you can assume that $bx_1x_2\cdots x_{m-1}<2^{31}$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer representing the denominator of the last fraction in the resulting Henry representation of Ahmes and Henry's algorithm for the input fraction $\frac{a}{b}$.

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

Sample Input	Output for the Sample Input
3 4 23 5 7 8 11	138 70 4070

Problem D
헨리
Time Limit: 1 Second

이제 10 살이 된 헨리(Henry)는 수학에 소질이 있다. 수학선생님인 아메스(Ahmes)는 오늘 헨리에게 분수에 대해 가르쳐줬고, 헨리는 분수를 이리저리 계산해보는 것이 너무 재미있었다. 그러던 중에 헨리는 1 보다 작은 분수를 여러 개의 서로 다른 단위분수의 합으로 표현할 수 있다는 것을 알아내었다. 여기서 단위분수란 분모가 1 인 분수를 말한다. 헨리는 여러 개의 분수에 대해 이를 시도해봤고, 시도해본 분수들은 모두 서로 다른 단위분수의 합으로 표현할 수 있었다. 예를 들어, $\frac{4}{23}$은 $\frac{1}{6}+\frac{1}{138}$와 같이 두 개의 단위 분수의 합으로 나타낼 수 있다.

헨리는 이런 발견을 선생님인 아메스에게 자랑스럽게 이야기했다. 아메스는 이를 듣고는 크게 기뻐하며 어린 제자를 칭찬했고, 이와 같이 하나의 분수를 서로 다른 단위분수의 합으로 표현한 것을 헨리식 표현법(Henry representation)이라고 이름 지었다. 즉, 분수 $\frac{a}{b}$의 헨리식 표현법은 총합이 $\frac{a}{b}$와 같게 되는 서로 다른 단위분수의 나열이다. 헨리와 아메스는 헨리식 표현법에 대하여 더욱 연구하였고, 마침내 모든 1 보다 작은 분수는 헨리식 표현법이 가능함을 증명하였다. 또한 헨리식 표현법이 유일하지 않다는 것도 알 수 있었다. 예를 들면, $\frac{5}{7}=\frac{1}{2}+\frac{1}{5}+\frac{1}{70}=\frac{1}{2}+\frac{1}{6}+\frac{1}{21}=\frac{1}{2}+\frac{1}{7}+\frac{1}{14}$와 같이 여러 가지 다른 헨리식 표현법이 존재할 수 있다. 단, 정의에 의해, 헨리식 표현법에는 같은 단위분수가 두 개 이상 포함될 수 없으므로 $\frac{2}{3}=\frac{1}{3}+\frac{1}{3}$는 헨리식 표현법이 아님을 유념해야 한다.

아메스와 헨리는 또한 주어진 분수의 헨리식 표현법을 구하는 간단한 방법도 고안해냈다. $a<b$ 인 양의 정수 a와 b로 이루어진 분수 $\frac{a}{b}$가 주어질 때에, 먼저 $\frac{1}{x_1}\le\frac{a}{b}$를 만족하는 가장 큰 단위 분수 $\frac{1}{x_1}$를 계산한다. 그런 다음 $\frac{a}{b}$ 에서 $\frac{1}{x_1}$를 뺀 나머지에 대하여 위의 과정을 반복한다. 즉, $\frac{1}{x_2}\le\frac{a}{b}-\frac{1}{x_1}$를 만족하는 가장 큰 단위분수 $\frac{1}{x_2}$를 계산하고 뺀다. 이러한 과정을 나머지가 남지 않을 때까지 반복한다. 그러면 서로 다른 단위분수들 $\frac{1}{x_1},\frac{1}{x_2},\frac{1}{x_3},\ldots$을 순서대로 얻게 되며 그들의 합은 정확히 $\frac{a}{b}$와 같아진다. 아메스와 헨리는 이 알고리즘이 항상 종료하며 합이 $\frac{a}{b}$가 되는 서로 다른 단위분수들, 즉 헨리식 표현법을 출력함을 증명하였다.

아메스와 헨리는 당신에게 그들의 알고리즘을 컴퓨터 프로그램으로 구현해줄 것을 부탁했다. 입력으로 주어지는 1 보다 작은 분수 $\frac{a}{b}$를 아메스와 헨리의 알고리즘을 수행하여 헨리식 표현법을 계산할 때에 마지막 단위 분수의 분모를 출력하는 프로그램을 작성하시오. 예를 들어. $\frac{a}{b}=\frac{5}{7}$라면, 아메스와 헨리의 알고리즘은 $\frac{5}{7}=\frac{1}{2}+\frac{1}{5}+\frac{1}{70}$을 출력하게 되므로 당신의 프로그램은 반드시 70 을 출력해야 한다.

입력(Input)

입력 데이터는 표준입력을 사용한다. 입력은 T 개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 테스트 데이터의 개수 T 가 정수로 주어진다. 각 테스트 데이터는 한 줄로 구성되며, 여기에는 입력 분수 $\frac{a}{b}$를 의미하는 두 개의 정수 a 와 b $(1 \leq a < b \leq 10{,}000)$ 가 주어진다. 이 때, a 와 b 는 서로 소이며, 입력분수 $\frac{a}{b}$에 대해 아메스와 헨리의 알고리즘을 실행했을 때에 출력되는 단위 분수가 순서대로 $\frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \dots, \frac{1}{x_m}$ 라면, $bx_1x_2 \cdots x_{m-1} < 2^{31}$의 부등식을 만족한다고 가정할 수 있다.

출력(Output)

출력은 표준출력을 사용한다. 각 테스트 데이터에 대해, 정확히 한 줄을 출력해야 하며 여기에는 정수 하나만을 출력한다. 이 정수는 아메스와 헨리의 알고리즘을 입력 분수 $\frac{a}{b}$에 대해 실행했을 때, 출력되는 헨리식 표현법의 마지막 단위분수의 분모와 같아야 한다.

다음은 세 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력예제(Sample Input)	출력예제(Output for the Sample Input)
3 4 23 5 7 8 11	138 70 4070

The 39th Annual ACM

International Collegiate Programming Contest

Asia Regional – Daejeon

Korea Nationwide Internet Competition

Problem E

Highway

Time Limit: 1 second

There is a country that has n cities. The country decides to build the longest straight highway connecting two cities. The highway may pass through some cities. That is, they want to find two cities that have maximum Euclidean distance over all pairs of the cities. The location of the cities is given by the integral coordinates of 2-dimensional plane.

For the case of following example, the two cities located in (12,0) and (-6,3) are the pairs that has maximum Euclidean distance.

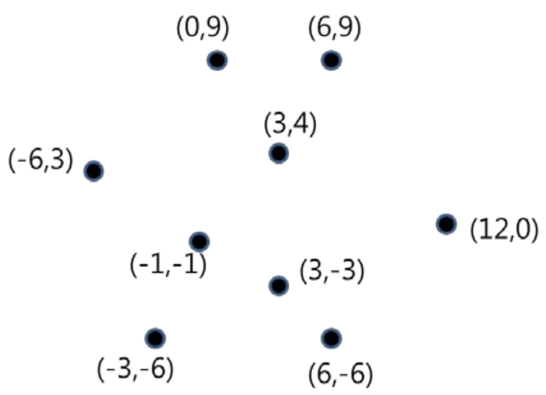


Figure 1. An example.

Write a program to find the location of two cities with the maximum Euclidean distance.

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 cities of the country, where $2 \leq n \leq 200{,}000$. Each of the following n lines contains a pair of integers x and y, representing the coordinate of a city. Note that all the coordinates of the cities are distinct, and $-10{,}000{,}000 \leq x, y \leq 10{,}000{,}000$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain four integers x_1, y_1, x_2, and y_2 that represent two city coordinates (x_1, y_1) and (x_2, y_2). If there are multiple such two cities, any one of them will be accepted.

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

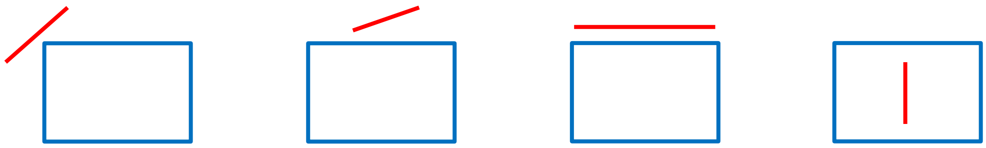
Sample Input	**Output for the Sample Input**
2 4 -100 -50 20 -50 -20 50 100 50 9 -1 -1 3 -3 6 -6 -3 -6 12 0 3 4 -6 3 0 9 6 9	-100 -50 100 50 -6 3 12 0

Problem F

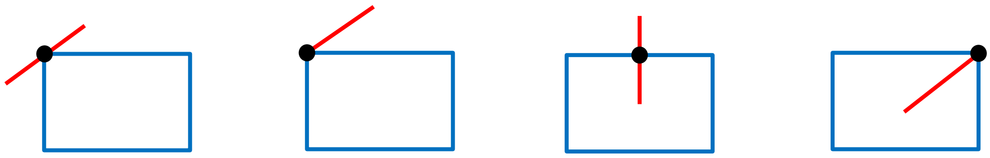
Intersections

Time Limit: 1 Second

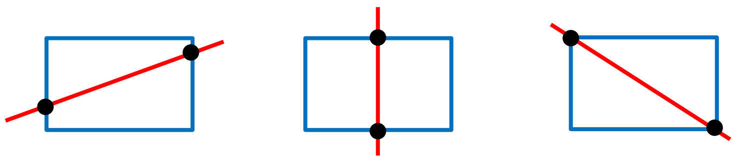
Your job is to write a program of solving a simple geometry problem for finding the number of intersection points of the boundary of a rectangle and a line segment. Each edge of a given rectangle is parallel to *x*-axis or *y*-axis. (*You are very lucky*!) The number of intersections between them is zero, one, two, or infinity. A case of 'infinity' occurs in a situation in which an edge of the rectangle and the segment are overlapped partially or wholly. See the figure below which shows examples of several situations between a rectangle and a segment.



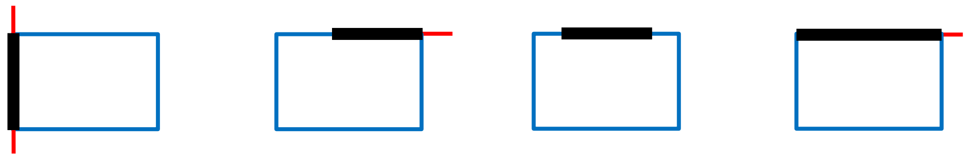
(a) Some examples of zero intersection between R and l.



(b) Some examples of one intersection between R and l.



(c) Some examples of two intersections between R and l.



(d) Some examples of infinite intersections between R and l.

Figure 1. Examples of several situations between a rectangle R(blue) and a line segment l(red).

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 four integers $\boldsymbol{xmin}$, $\boldsymbol{ymin}$, $\boldsymbol{xmax}$, and $\boldsymbol{ymax}$ representing a rectangle R, where $(\boldsymbol{xmin}, \boldsymbol{ymin})$ and $(\boldsymbol{xmax}, \boldsymbol{ymax})$ represent the coordinates of the lower left corner and upper right corner of *R*, respectively, $-10,000 \le xmin < xmax \le 10,000$ and $-10,000 \le ymin < ymax \le 10,000$. The next line contains four integers x_1, y_1, x_2, and y_2 representing a line segment l, where (x_1, y_1) and (x_2, y_2) represent the coordinates of two end points of l, respectively, $-10,000 \le x_1,\ y_1, x_2, y_2 \le 10,000$, and the length of l is greater than zero.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer representing the number of the intersections of the boundary of a rectangle and a line segment given by input. If the number of the intersections is infinity, then your program should output “4” instead.

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

Sample Input

```
16
0 0 8 4
2 6 -2 3
0 0 8 4
0 4 9 4
0 0 8 4
3 5 6 6
0 0 8 4
-2 5 10 -1
0 0 8 4
0 5 8 5
0 0 8 4
4 3 4 1
0 0 8 4
-2 3 2 5
0 0 8 4
2 4 6 4
0 0 8 4
0 4 4 7
0 0 8 4
4 2 4 4
0 0 8 4
4 2 8 4
0 0 8 4
0 2 3 4
0 0 8 4
-4 0 12 4
0 0 8 4
4 8 4 -1
0 0 8 4
0 -2 0 6
0 0 8 4
3 4 10 4
```

Output for the Sample Input

```
0
4
0
2
0
0
1
4
1
1
1
2
2
2
4
4
```

Problem G
Mutation
Time Limit: 0.2 seconds

The human DNA sequence can be considered as a long string made up with four letters, A, G, T, and C. It is known that some diseases are related to a marker, a short substring of the DNA sequence. If your DNA contains it, then it is likely that you have a higher probability of suffering from those diseases.

Sometimes it is not enough to find markers from your DNA sequences. Mutation can change the marker. That is, the marker can be divided into three parts and the middle one can be reversed. Note that the first or the third one can be an empty string. For example, if the marker is AGGT, considering the mutation, there can be six cases: GAGT, GGAT, TGGA, AGGT, ATGG, and AGTG.

Given your DNA sequence and the marker, write a program which computes the number of occurrences of the marker or its mutations. For example, if your DNA sequence is ATGGAT and the marker is AGGT again, the answer is 3 as it includes ATGG, TGGA, and GGAT.

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 integers $\boldsymbol{n}$ and $\boldsymbol{m}$, the length of your DNA sequence and that of the marker, respectively, where $\mathbf{1} \le \boldsymbol{n} \le \mathbf{1,000,000}$ and $\mathbf{1} \le \boldsymbol{m} \le \mathbf{100}$. The second line of each test case contains your DNA sequence. The third line of each test case contains the marker. All the strings are made up with four letters, A, G, T, and C.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer which is the number of occurrences of the marker and its mutated strings in your DNA sequence. If there is no such occurrence, print 0.

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

Sample Input	Output for the Sample Input
2 6 4 ATGGAT AGGT 6 4 ATGGAT AGCT	3 0

Problem H
Registration
Time Limit: 1 second

Print out your ICPC team number and team name.

Input

No input is given for this problem.

Output

Your program is to write to standard output. Print exactly two lines. The first line should contain your team number, and the second line should contain your full team name, even if your team name contains one or more blanks (a character of ASCII 32).

The following shows sample input and output, where the team number is `123` and the team name is `Your_ICPC_Team_Name`. Notice that no input is given.

Sample Input	Output for the Sample Input
	123 Your_ICPC_Team_Name

Problem H

등록

Time Limit: 1 second

자신의 ICPC 팀 번호와 팀 이름(team name)을 그대로 출력하는 프로그램을 작성하시오.

Input

이 문제는 입력이 없다.

Output

표준출력(standard output)으로 출력해야 한다. 첫 줄에 자신의 팀 번호, 둘째 줄에 팀 이름을 출력한다. 출력할 팀 이름은 공백문자(ASCII 코드 32 번인 문자)를 포함하더라도, 공백문자를 포함하여 완전한 이름을 출력해야 한다.

다음은 팀 번호가 123 번, 팀 이름(team name)이 Your_ICPC_Team_Name 인 경우의 입출력 예제이다. 참고로 입력이 없는 것에 주의한다.

Sample Input	Output for the Sample Input
	123 Your_ICPC_Team_Name

Problem I
Stains
Time Limit: 1 Second

During the ChuSeok holidays you had quite a few guests, including some very young boys. After they went back to their respective homes, you found some stains on your bed. The stains are deep and actually go into the mattress. It is quite expensive to clean the whole mattress, so you have decided to buy a few of the clean-mattress-squares, which can clean some area of the mattress. The surface of the mattress can be divided into an $m \times n$ grid, i.e., a rectangle with m unit squares vertically and n unit squares horizontally. A coordinate system is defined so that the unit square at the top-left corner has coordinate $(1, 1)$ and the unit square at the bottom-right corner has (m, n). In each square there may be a stain. Each of your clean-mattress-squares is able to clean up all stains in a 3×3 area. Your mission is to minimize the number of clean-mattress-squares to be used while, of course, removing all of the stains.

Figure 1 is an example where the total mattress is a 6×11 grid and there are four stains located at unit squares whose coordinates are $(4, 3)$, $(6, 5)$, $(3, 6)$, and $(4, 7)$, respectively. In this case, two clean-mattress-squares will suffice.

Given the size of the mattress and the coordinates of the unit squares where stains are located, you are to write a program that computes the minimum number of clean-mattress-squares to be used.

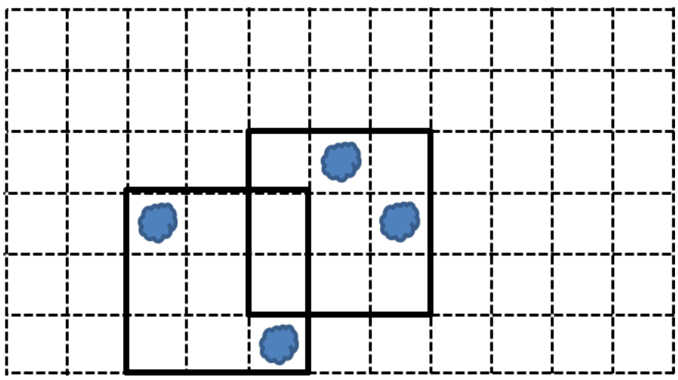


Figure 1

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 integers m and n, giving the vertical and horizontal size of the mattress, respectively, where $3 \leq m \leq 10$ and $3 \leq n \leq 1{,}000$. In the next line, a nonnegative integer c is given, indicating the number of unit squares where stains are located. In the next c lines, the coordinates of the stains are given one per line. It always holds that $0 \leq c \leq mn$. Also, the coordinates given are all different and are all on the mattress.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain a nonnegative integer, which is the minimum number of clean-mattress-squares needed to remove all of the stains.

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

Sample Input	Output for the Sample Input
2 6 11 4	2 2

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

Problem J
Switch Array
Time Limit: 1 second

We have a locking system with a linear switch array $s_1 s_2 s_3 \ldots s_{i-1}\ s_i$ where s_i is a unit switch. Each unit switch has one of two exclusive states { 0, 1 }. In this problem $s_i = x$ denotes the unit switch s_i which has a state $x \in \{0,1\}$ exclusively. The following Table shows a state of one switch array with 8 units.

switch	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
state	0	1	1	0	1	1	0	0

Assume that the number of unit switches is N. You can control the unit switch s_i according to the following two operation rules. Note that only one switch can be toggled at a time.

Operation Rule 1) You can toggle the rightmost switch s_N at any time.
Operation Rule 2) You can toggle the switch s_i only if s_{i+1} =1 and all its right switches are 0, that is $s_{i+2} s_{i+3} \ldots s_{N-1}\ s_N$ = 000…00. This rule works if $i + 1 = N$. This means that Rule 2 works even if $s_{i+2} s_{i+3} \ldots s_{N-1} s_N$ does not exist.

We can unlock this system if we set $s_i = 0$ for all $i, 1 \leq i \leq N$. For a switch array given, you are required to unlock this system by manipulating $s_i = 0$ with two operations explained above. The following table shows how to unlock a switch array with the state '1111'. In this case you can unlock this system using 10 operations which is the minimum operations to unlock. We want to know the minimal number of switch operation to unlock for a switch array given.

operation	s_1	s_2	s_3	s_4
0	1	1	1	1
1	1	1	0	1
2	1	1	0	0
3	0	1	0	0
4	0	1	0	1
5	0	1	1	1
6	0	1	1	0
7	0	0	1	0
8	0	0	1	1
9	0	0	0	1
10	0	0	0	0

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 one binary string *B* to denote the switch array given. The first(last) character of B is s_1 (s_N). The length of *B* is bounded by $\mathbf{2} \leq |\mathbf{B}| \leq \mathbf{31}$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the minimum of operations to unlock the switch array with *B* state.

The following shows sample input and output for 5 test cases.

Sample Input	Output for the Sample Input
5 1111 11111 1010101010 000 000000010	10 21 819 0 3

The 39th Annual ACM

International Collegiate Programming Contest

Asia Regional – Daejeon

Korea Nationwide Internet Competition

Problem K

Traffic Jams

Time Limit: 5 seconds

You are in charge of designing an advanced centralized traffic management system for smart cars. The goal is to use global information to instruct morning commuters, who must drive downtown from the suburbs, how best to get to the city center while avoiding traffic jams.

Unfortunately, since commuters know the city and are selfish, you cannot simply tell them to travel routes that take longer than normal (otherwise they will just ignore your directions). You can only convince them to change to different routes that are equally fast.

The city's network of roads consists of intersections that are connected by bidirectional roads of various travel times. For some pair of two intersections, there can be several roads which connect these two intersections. Each commuter starts at some intersection, which may vary from commuter to commuter. All commuters end their journeys at the same place, which is downtown at intersection 1. If two commuters attempt to start travelling along the same road in the same direction at the same time, there will be congestion; you must avoid this. However, it is fine if two commuters pass through the same intersection simultaneously or if they take the same road starting at different times.

Determine the maximum number of commuters who can drive downtown without congestion, subject to all commuters starting their journeys at exactly the same time and without any of them taking a suboptimal route.

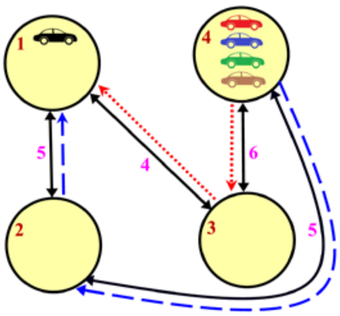


Figure 1: Illustration of Sample Input 2.

In Figure 1, cars are shown in their original locations. One car is already downtown. Of the cars at intersection 4, one can go along the dotted route through intersection 3, and another along the dashed route through intersection 2. But the remaining two cars cannot reach downtown while avoiding congestion. So a maximum of 3 cars can reach downtown with no congestion.

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 three integers n, m, and c, where n $(1 \le n \le 25{,}000)$ is the number of intersections, m $(0 \le m \le 50{,}000)$ is the number of roads, and c $(0 \le c \le 1{,}000)$ is the number of commuters. Each of the next m lines contains three integers x_i, y_i, and t_i describing one road, where x_i and y_i $(1 \le x_i,\ y_i \le n)$ are the distinct intersections which the road connects, and t_i $(1 \leq t_i \leq 10{,}000)$ is the time it takes to travel along that road in either direction. You may assume that downtown is reachable from every intersection. The last line contains c integers listing the starting intersections of the commuters.

Output

Print the maximum number of commuters who can reach downtown without congestion.

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

Sample Input	Output for the Sample Input
2 3 3 3 1 2 42 2 3 1 2 3 1 2 3 3 4 4 5 1 2 5 1 3 4 4 2 5 4 3 6 4 4 4 4 1	2 3