2015
서울대학교 교내 프로그래밍 경시대회

주최 및 주관

후원

ESTsoft

DEVSISTERS

2015년 9월 13일

참가자를 위한 도움말

주의 사항

- 대회 시간은 13:00부터 17:00까지입니다. 대회가 진행되는 동안 인터넷 검색 및 전자기기 사용 등을 하실 수 없습니다. 단, 아래의 문서에 한해 대회 진행 중에도 참고하실 수 있으며, 책과 노트 등을 가져오신 경우 역시 참고하실 수 있습니다.
  - STL documentation: http://www.sgi.com/tech/stl/
  - JDK documentation: http://docs.oracle.com/javase/8/docs/api/
  - C++ reference: http://en.cppreference.com/w/cpp
- 대회 진행은 PC² 프로그램을 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 모든 입력은 *표준 입력*으로 주어지며, 모든 출력은 *표준 출력*으로 합니다.
- 테스트 케이스가 존재하는 문제의 경우, 테스트 케이스에 대한 출력을 모아서 하실 필요 없이, 각 테스트 케이스를 처리할 때마다 출력해도 괜찮습니다.
- *중요!! 리턴 코드와 표준 오류 (standard error, stderr) 스트림 출력* 에 주의하십시오. 프로세스가 0 이 아닌 리턴 코드를 되돌리는 경우나 *표준 오류 스트림에 출력*을 하는 경우 "*런타임 에러*" 를 받게 됩니다.
- 문제에 대한 질의 사항은 PC² 의 'Request Clarification' 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 "*No response, read problem statement carefully*" 로 대답될 수 있으므로 유의하십시오.

채점 결과에 대하여

Yes 제출하신 답안이 모든 테스트 데이터를 정해진 시간 안에 통과하여 정답으로 인정되었음을 의미합니다.

No - Compilation Error 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

No - Run-time Error 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

No - Time-limit Exceeded 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다. 시간 제한은 공개되지 않습니다.

No - Wrong Answer 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

No - Excessive Output 제출하신 답안 프로그램이 지나치게 많은 양의 출력물을 생성하여 강제로 종료되었음을 의미합니다.

No - Output Format Error 제출하신 답안 프로그램이 정해진 출력 형식을 따르지 않았음을 의미합니다.

만약 여러 가지의 원인으로 인해 "*Yes*" 가 아닌 다른 결과를 얻으셨다면, 그 중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 출력 형식도 잘못된 코드를 제출하신 경우 대부분 "*No - Output Format Error*" 를 받으시게 되지만, 경우에 따라서 "*No - Wrong Answer*" 를 받을 수도 있습니다.

Problem A. 까만 선글라스

EDM을 좋아하는 G.Park은 최대한 많은 사람들에게 EDM을 전파하려고 한다. 그는 EDM이 최고의 음악 장르라고 얘기하며, 모든 곡을 EDM으로 바꿀 수 있다고 주장한다. EDM을 어려워하는 사람들을 위해, 그는 다음과 같은 멋진 법칙을 고안해냈다.

"가사의 첫 음절을 여러 번 반복하면 곧 EDM이 된다."

즉, "까만 선글라스"라는 가사의 첫 음절을 5번 반복한 "까까까까까만 선글라스"는 훌륭한 EDM 버전의 가사가 된다. 그는 이러한 규칙을 적용하여 많은 곡을 EDM으로 편곡하고 싶어했는데, 세상에 알려진 수많은 곡을 손수 바꾸는 것은 매우 힘이 드는 일이기 때문에, 당신에게 자동으로 EDM 버전의 가사를 만드는 프로그램을 만들어달라고 부탁했다. 당신은 세상에 EDM을 널리 알리려는 G.Park의 꿈을 이뤄줄 수 있을까?

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스는 띄어쓰기가 없는 가사 S와 반복 횟수 $N(1 \leq N \leq 100)$으로 구성된다. 가사 S는 알파벳 소문자로 구성되는데, 여기에서 첫 모음('a', 'e', 'i', 'o', 'u')이 등장할 때까지의 부분을 첫 음절로 간주한다. 모든 가사에는 하나 이상의 모음이 들어간다.

Output

각 테스트 케이스마다 EDM 버전의 가사를 한 줄씩 출력한다.

Sample input and output

standard input	standard output
3 kamansunglass 5 zum 4 devsisters 2	kakakakakamansunglass zuzuzuzum dedevsisters

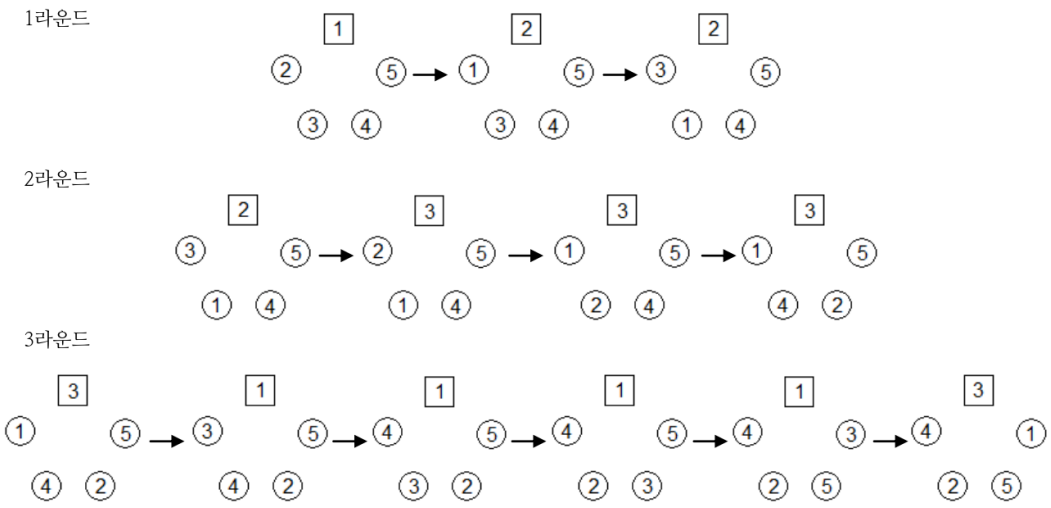
Problem B. 다트

비공개

Problem C. 쿠키 옮기기 (외부 기출)

쿠키 옮기기는 N 개의 쿠키를 이용하여 할 수 있는 게임이다. 각각의 쿠키에는 딸기맛 쿠키, 커피맛 쿠키, 박하사탕맛 쿠키와 같은 이름이 있지만 이 문제에서는 편의를 위해 1번, 2번, ..., N 번 쿠키라고 부르자. 쿠키는 원형으로 배치되어 있으며, 쿠키가 있는 자리 중 하나는 사각형, 다른 곳은 원형으로 위치를 표시한다. 1번 쿠키는 사각형 영역에 배치하고, 다른 쿠키는 번호 순서대로 원형 영역에 각각 배치한다.

게임은 K 번의 라운드로 진행된다. 각각의 라운드는 사각형 영역에 있는 쿠키를 시계 반대 방향으로 인접한 쿠키와 위치를 바꾸는 과정을 반복하며 이루어진다. i 번째 라운드에서 위치를 바꾸는 횟수는 p_k 번으로, 이 때 p_k 는 k 번째 소수를 의미한다. 예를 들어, $N = 5$ 이고 $K = 3$ 일 때의 라운드 진행은 아래와 같다.



N, K, A 가 주어졌을 때, 모든 라운드가 끝난 후 A 번 쿠키의 양 옆에 위치한 쿠키의 번호를 출력하는 프로그램을 작성하시오.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스는 한 줄로 구성되며, 세 개의 정수 N, K, A ($3 \le N \le 5,000,000$, $1 \le K \le 500,000$, $1 \le A \le N$) 가 차례로 입력된다.

Output

각 테스트 케이스마다, 모든 라운드가 끝난 후 A 번 쿠키의 오른쪽에 있는 쿠키의 번호와 왼쪽에 있는 쿠키의 번호를 차례로 출력한다.

Sample input and output

standard input	standard output
3 5 3 1 5 3 2 5 4 5	3 5 5 4 3 2

Notes

Croatian Olympiad in Informatics 2010 #2

Problem D. Pork barrel (외부 기출)

Winning the election was simpler than you expected: it was enough to promise to finally build a good quality, country-wide road infrastructure, of course without crippling the budget... Your happiness did not last long, however: it seems, that the citizens have found a way to actually hold you accountable for your promise!

There are n major cities in your country. The Ministry of Transport has prepared a detailed map, outlining m possible highway connections, together with their costs. The Quality Assurance Committee will not let you build a highway cheaper than l, and the National Spendings Regulatory Committee will not let you build a highway more expensive than h. To claim a "country-wide" network, you have to connect (possibly indirectly) as many pairs of cities, as it is possible within these two constraints. You have to find the cheapest way to do it, and you have to find it quickly! Of all networks that meet the constraints and connect the most pairs of cities, compute the cost of the cheapest one.

To make things worse, both committees are heavily influenced by your angry competitors: each time you publish your hard-prepared plan, they immediately change their rulings l and h, and you are forced to start from scratch.

Input

The first line of input contains the number of test cases T. The descriptions of the test cases follow:

The first line of each test case contains integers N and M ($1 \leq N \leq 1,000$; $0 \leq M \leq 100,000$) - the number of cities in the country, and of possible direct connections, respectively.

Each of the following M lines contains three integers X, Y, W ($1 \leq X \neq Y \leq N$; $1 \leq W \leq 1,000,000$), denoting that the cities X and Y can be connected by a bidirectional highway at cost W. There might be many ways to connect a single pair of cities.

The following line contains an integer Q ($1 \leq Q \leq 1,000,000$) - the number of rulings of the committees. Each of the following Q lines contains two integers. The first of the lines contains the initial rulings L_1, H_1, given directly. The rest of the rulings are encoded. The numbers in the jth of the lines for $j > 1$ are $L_j + C_{j-1}$ and $H_j + C_{j-1}$, where L_j and H_j are the actual rulings and C_{j-1} is the correct answer for the preceding rulings L_{j-1} and H_{j-1}.

All rulings satisfy $1 \leq L_j \leq H_j \leq 1,000,000$.

Output

For each test case, output Q lines, one for each ruling. In the jth of them, output the minimal cost C_j of building a highway network which adheres to the committees' constraints, and creates the maximum number of connected pairs of cities.

Sample input and output

standard input	standard output
1 5 7 1 2 2 2 3 4 3 4 3 4 5 1 5 1 3 2 5 4 1 4 5 5 1 2 4 7 11 12 11 13 18 19	3 9 8 14 13

Notes

The actual rulings of the committees are $(1,2)$, $(1,4)$, $(2,3)$, $(3,5)$ and $(4,5)$. The cheapest highway networks adhering to these rulings consist of connections $\{(1,2),(4,5)\}$, $\{(2,1),(1,5),(5,4),(4,3)\}$, $\{(1,2),(1,5),(3,4)\}$, $\{(1,5),(5,2),(2,3),(3,4)\}$ and $\{(3,2),(2,5),(1,4)\}$, respectively.

출처: CERC 2014

Problem E. 위젯 배치 (외부 기출)

zum은 더 편리한 인터넷을 지향하는 인터넷 포털로, 여러 위젯을 배치하여 사용자가 원하는 방향으로 페이지를 구성할 수 있다는 점이 큰 특징 중 하나이다. 브라이언은 여러 위젯을 사용하면서 최대한 편리한 페이지를 만들어보기 위해 다음과 같은 실험을 해보기로 했다.

zum은 원하는 만큼 페이지를 추가하여 사용할 수 있는데, 총 M 개의 페이지가 있다고 가정하자. 각 페이지의 번호는 1 번부터 M 번까지 차례로 붙어 있다. 처음에는 모든 페이지가 비어있는 상태인데, 여기에 N 개의 위젯을 하루에 하나씩 추가해보려고 한다.

한 페이지에 지나치게 많은 위젯이 배치되어 있는 경우 피로감을 유발할 수 있기 때문에, 브라이언은 위젯을 추가할 때마다 다음과 같이 '피로도'를 계산해보기로 했다. '피로도'는 위젯을 추가할 당시 해당 페이지에 있는 위젯의 개수이다. 즉, 2 개의 위젯이 있는 페이지에 새로운 위젯을 추가하면 이 페이지의 피로도는 3 이 된다. 피로도가 지나치게 높은 페이지는 사용하기 불편하기 때문에, 가끔씩 페이지에 배치된 위젯을 모두 삭제하는 절차를 추가하기로 했다. 하지만 이는 매우 귀찮은 일이므로 실험이 끝날 때까지 최대 K 번만 모든 위젯 삭제를 진행하기로 했다. 위젯을 하나씩 추가할 때마다 페이지의 피로도를 기록하고, 실험이 끝난 뒤 모든 피로도를 합한 값이 최소가 되도록 하는 것이 실험의 목표이다.

실험을 하기에 앞서, 브라이언은 위젯을 배치할 순서와 페이지를 미리 정한 뒤 하나씩 추가하기로 했다. 하지만 일일이 위젯을 추가하고 삭제하는 과정을 진행하는 것은 매우 귀찮은 일이기 때문에, 당신에게 실험을 대신 해주는 프로그램을 부탁하게 되었다. 총 N 개의 위젯을 배치하는 과정에서 계산한 피로도의 합을 최소화하는 프로그램을 작성하라.

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 배치할 위젯의 수 N ($1 \leq N \leq 1,000,000$)과 위젯을 배치할 페이지의 수 M ($1 \leq M \leq 100$), 가능한 '모든 위젯 삭제'의 횟수 K ($1 \leq K \leq 500$)가 주어진다.

이어지는 N줄에는 한 줄에 하나씩 정수 W_i ($1 \leq i \leq N$)가 주어지며, 이는 i번째 위젯을 배치할 페이지의 번호를 나타낸다. ($1 \leq W_i \leq M$)

Output

각 테스트 케이스에 대해 피로도의 최소값을 각각 한 줄씩 출력한다.

Sample input and output

standard input	standard output
2 5 1 2 1 1 1 1 1 11 2 3 1 2 1 2 1 2 1 2 1 2 1	7 18

Notes

첫 번째 예제에 대한 설명: 최대 두 번 모두 삭제를 진행할 수 있는데, 첫 번째 위젯을 배치한 후와 세 번째 위젯을 배치한 후 모두 삭제를 진행하면 각 단계에서의 피로도는 1, 1, 2, 1, 2가 되어 합이 7이 된다. 모두 삭제를 한 번도 진행하지 않을 경우 각 단계에서의 피로도는 1, 2, 3, 4, 5가 될 것이다.

두 번째 예제에 대한 설명: 다음은 여러 가지 가능한 방법 중 하나이다. 세 번째 위젯과 일곱 번째 위젯을 첫 페이지에 배치한 직후에 위젯을 모두 삭제하고, 여섯 번째 위젯을 두 번째 페이지에 배치한 직후 위젯을 모두 삭제한다. 그러면 각 단계에서의 피로도는 1, 1, 2, 2, 1, 3, 2, 1, 1, 2, 2가 된다.

출처: COCI 2014/2015 #1 ZABAVA

Problem F. Matrix Game (외부 기출)

Two players A and B play the following game.

1. First, a matrix M of size $N * M$ is chosen, and filled with non-zero numbers.
2. Player A starts the game and the players play alternately.
3. In his turn, a player chooses any row which has atleast one non zero nuumber in it. In this row, the leftmost non zero number is chosen. Let this number be K. The player subtracts any number between 1 and K inclusive from it.
4. The game ends when all the numbers in the matrix M are 0.
5. The player who plays last wins the game.

Given N, M and the initial matrix, determine who wins the game. Assume that both players play optimally.

Input

The first line contains the number of test cases T ($T \leq 1,000$).

Each test case consists of 2 numbers N and M ($1 \leq N, M \leq 50$). There follow N lines each having M integers. The jth number on the ith line is the number $M[i][j]$. The initial matrix values are between 1 and 50 inclusive. There is a blank line between consecutive test cases.

Output

Output T lines, one for each case. Output "FIRST" if player A wins, else output "SECOND".

Sample input and output

standard input	standard output
3 2 2 1 1 1 1 1 3 2 1 1 2 2 3 2 3 2	SECOND FIRST SECOND

Notes

출처: Codechef Snackdown

Problem G. Eclipse

2015년 3월 20일은 북유럽 등지에서 개기일식이 있었던 날이다. 개기일식은 1년에 한두 번 꼴로 일부 지역에서만 짧게 일어나는 드문 일이기 때문에 일식이 일어나는 장소에 가서 직접 그 놀라운 광경을 목격하려는 사람이 많다. 브라이언은 개기일식을 보기 위해 북유럽에 있는 작은 섬인 페로 제도에 힘들여 갔지만 결국 날씨 때문에 그 광경을 못 봤다는 안타까운 전설이 있다.

일식이 일어났을 때 달의 위치에 따라 일식의 종류가 나뉘는데, 달이 완전히 해를 가리는 경우는 개기 일식, 달의 각크기가 해보다 작아 달이 해 안에 들어가는 경우를 금환 일식으로 부른다. 즉, 지구에서 본 달의 크기가 해의 크기보다 큰지 작은지를 비교하면 개기일식이 일어날 수 있는지를 알 수 있다.

아래의 사진은 일식이 일어나는 과정에 찍힌 사진이다. 브라이언은 이 사진을 보고 개기일식이 일어날 수 있는 상황인지를 알고 싶어졌다. 브라이언은 복잡한 계산을 하고 싶지 않아서 태양과 달은 완전한 원형을 이루고, 일식이 일어나는 중에 해와 달의 크기가 변하는 일이 없다고 가정했다. 이렇게 가정할 경우 개기일식이 일어나는지를 계산할 수 있다는 사실을 알게 되었다. 사진에 찍힌 해와 달의 좌표 정보가 주어졌을 때, 개기일식이 일어날 수 있는 상황인지를 판단하는 프로그램을 작성하라.

[1]

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스는 8개의 정수 좌표 A_x, A_y, B_x, B_y, C_x, C_y, D_x, D_y로 구성된다. A와 C는 해와 달의 테두리가 맞닿는 점의 좌표이며, B는 해의 테두리 상의 좌표, D는 달의 테두리 상의 좌표이다. 네 개의 점 중 겹치는 점은 없으며, 모든 좌표의 절대값은 $1,000$ 이하이다. 해와 달의 반지름은 10^5 이하이고, 해와 달의 반지름의 오차는 항상 10^{-5} 이상이다.

Output

각 테스트 케이스마다 개기일식이 가능한지를 출력한다. 가능한 경우 "YES"를, 불가능한 경우 "NO"를

[1] original image from http://www.sciencetimes.com/articles/875/20141019/taking-a-bite-out-of-sunlight-us-expects-a-partial-solar-eclipse.htm

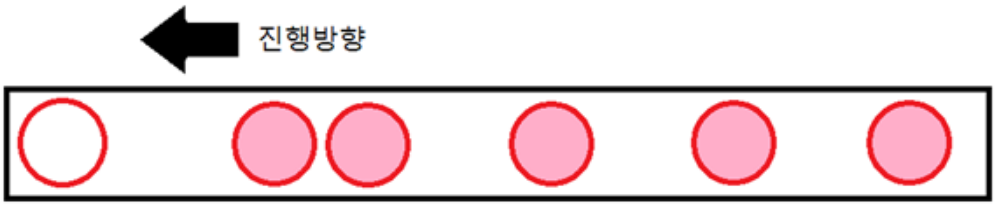
출력한다.

Sample input and output

standard input	standard output
2 -5 -5 7 5 -1 3 0 0 2 7 11 5 4 12 3 9	NO YES

Problem H. 북의 달인

최근에 SNUPS에서 북의 달인 게임을 개발하였다.



게임의 한 장면. 속이 찬 동그라미가 일정한 속도로 왼쪽으로 이동한다. 속이 빈 동그라미와 겹치는 순간에 북을 두드리면 된다.

타이밍을 맞추어 북을 두드리면 점수를 얻을 수 있다. 그러한 타이밍을 나타내는 것을 "노트"라 부르기로 하자. 게임은 이러한 노트들이 여러 개 모여 이뤄진다. 북의 달인 게임에서는 정해진 시간의 앞뒤 50 ms 사이에 한 번이라도 북을 두드리면 해당 "노트"를 올바르게 쳤다고 인정해준다. 즉, x ms에 쳐야 하는 "노트"라면 $x-50$ ms부터 $x+50$ ms 사이에 북을 두드리면 그 노트를 올바르게 친 것으로 인정해준다. 정확히 $x-50$ ms 또는 정확히 $x+50$ ms에 두드린 것도 올바르게 친 것으로 인정해준다.

북의 달인 게임에는 콤보 시스템이 있다. 연속으로 노트를 올바르게 친 경우 콤보수가 높아진다. 게임이 시작할 때 콤보수는 0이다. 노트를 올바르게 친 경우 콤보수가 1 증가한다. 만약 중간에 한 번이라도 올바르게 치지 못하게 되면 콤보수는 그 순간 0으로 초기화 된다. 게임이 종료되면 플레이 도중 도달한 콤보수의 최대값이 표시된다.

북의 달인을 즐기는 찬민이는 최대 콤보수를 높이고 싶어한다. 그런데 슬프게도 찬민이의 손은 느려서 P ms 미만의 간격으로 연타하는 것이 불가능하다. 즉, 가장 마지막에 노트를 친 시간이 x ms였다면, 다음 노트는 아무리 빠르게 쳐도 $x+P$ ms (inclusive)는 되어야 칠 수 있다. 이 때 찬민이가 달성할 수 있는 최대 콤보수는 얼마일까?

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스의 첫 줄에 N과 P가 공백을 사이에 두고 주어진다. ($1 \le N \le 1,000,000$, $100 \le P \le 1,000,000$)

두 번째 줄에는 노트의 타이밍을 의미하는 N개의 정수 x_i가 작은 수부터 순서대로 들어있다. ($100 \le x_i \le 2 * 10^9$) $101 \le x_{j+1} - x_j$ ($1 \le j \le N-1$)이 보장된다.

Output

찬민이가 얻을 수 있는 최대 콤보수를 한 줄에 하나씩 출력한다.

Sample input and output

standard input	standard output
3 4 250 100 300 500 700 6 250 100 300 700 900 1100 1300 6 100 100 201 302 403 504 605	3 5 6

Problem I. 데이터 만들기

찬민이는 대회를 위해 쉬운 문제를 만들기로 했다. 다음은 찬민이가 생각한 문제이다.

> 1 이상의 정수로 만들어진 길이 N의 수열이 주어진다. 이 때 질의가 Q개 주어진다. 각각의 질의는 a번째 수부터 b번째 수까지 다 더하면 얼마인지 구하라는 것이다. 단, $1 \leq a \leq b \leq N$이다. 프로그램은 이 질의에 답하면 된다.

쉬운 문제 입력 예

10	N
1 2 3 4 5 6 1 2 3 4	수열
3	Q
3 3	3번째 수부터 3번째 수까지의 합은?
1 4	1번째 수부터 4번째 수까지의 합은?
6 8	6번째 수부터 8번째 수까지의 합은?

쉬운 문제 출력 예

```
3
10
9
```

특이하게도 찬민이는 입력 수열을 만들지도 않고 질의와 출력데이터부터 만들었다. 이제 입력 수열을 만들 차례다. 찬민이는 무의미하게 입력 데이터를 키우고 싶지는 않았기 때문에, 수열의 원소의 최대값을 가능한 작게 만들고 싶다. 수열의 원소의 최대값을 가능한 작게 만들었을 때, 수열의 원소의 최대값은 얼마인지 구하여라.

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

테스트 케이스의 첫 번째 줄에는 N, Q $(1 \leq N, Q \leq 200)$가 공백을 사이에 두고 차례로 주어진다. N은 수열의 길이를 의미하고, Q는 질의의 개수를 의미한다.

테스트 케이스의 두 번째 줄부터 Q개의 줄에는 질의의 정보와 출력을 의미하는 A, B, C $(1 \leq A \leq B \leq N, 1 \leq C \leq 10,000)$가 공백을 사이에 두고 차례로 주어진다. A와 B는 질의의 시작 위치와 끝 위치를 나타낸다. C는 해당 질의에 대한 답을 의미한다.

Output

각 테스트 케이스마다 수열의 원소의 최대값을 가장 작게 했을 때의 값을 한 줄에 출력한다. 만약 불가능하다면 -1을 출력한다.

Sample input and output

standard input	standard output
3 1 1 1 1 42 1 2 1 1 1 1 1 2 10 3 3 3 3 1 4 10 6 8 9	42 -1 3

Problem J. 까마귀

2차원 평면 위에는 까마귀가 한 마리 살고 있다. 까마귀는 오늘 하루도 먹이를 찾기 위해, 반짝이는 것을 찾기 위해 많이 돌아다녔다.

평면 상에서 $y < 0$인 부분은 땅이기 때문에 까마귀가 갈 수 없는 곳이다. 또한 하나의 산이 있어서 산의 안쪽에도 까마귀는 갈 수 없다. 산은 $P_1(x_1, y_1)$에서 $P_N(x_N, y_N)$까지의 N개의 정점으로 나타낼 수 있다. x_i는 증가하는 순서이고, $y_i \geq 0$이며 $y_1 = y_N = 0$이다. P_1에서 P_N까지를 순서대로 연결하고, P_N과 P_1을 연결한 다각형이 산이 된다. 까마귀는 산의 변 위로는 올라갈 수 있지만, 내부로는 들어갈 수 없다.

까마귀는 오늘 하루 $M-1$번 이동했다. 처음에는 $Q_1(X_1, Y_1)$에 위치해 있었으며, 다음으로는 $Q_2(X_2, Y_2)$, $\cdots$, 마지막으로 $Q_M(X_M, Y_M)$를 방문하여 하루를 끝냈다. 까마귀는 똑똑하기 때문에 Q_i에서 Q_{i+1}로 이동할 때 땅의 내부나 산의 내부로 이동하지 않으면서 거리가 가장 짧은 경로로 이동하였다. 오늘 하루 까마귀가 이동한 거리는 얼마인지 구하는 프로그램을 작성하라.

Input

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스 마다 첫 번째 줄에는 $N(3 \leq N \leq 2 * 10^5)$이 주어진다. 다음 N개의 줄의 i번째 줄에는 정수 좌표 x_i, $y_i(-10^8 \leq x_i \leq 10^8, 0 \leq y_i \leq 10^8)$가 공백 하나로 구분되어 주어진다. $x_i < x_{i+1}(1 \leq i < N)$와 $y_1 = y_N = 0$을 만족한다.

다음 줄에는 $M(2 \leq M \leq 2 * 10^5)$이 주어진다. 다음 M개의 줄의 i번째 줄에는 X_i, $Y_i(-10^8 \leq X_i \leq 10^8, 0 \leq Y_i \leq 10^8)$가 공백 하나로 구분되어 주어진다. (X_i, Y_i)가 산 아래에 위치해 있는 경우는 없다.

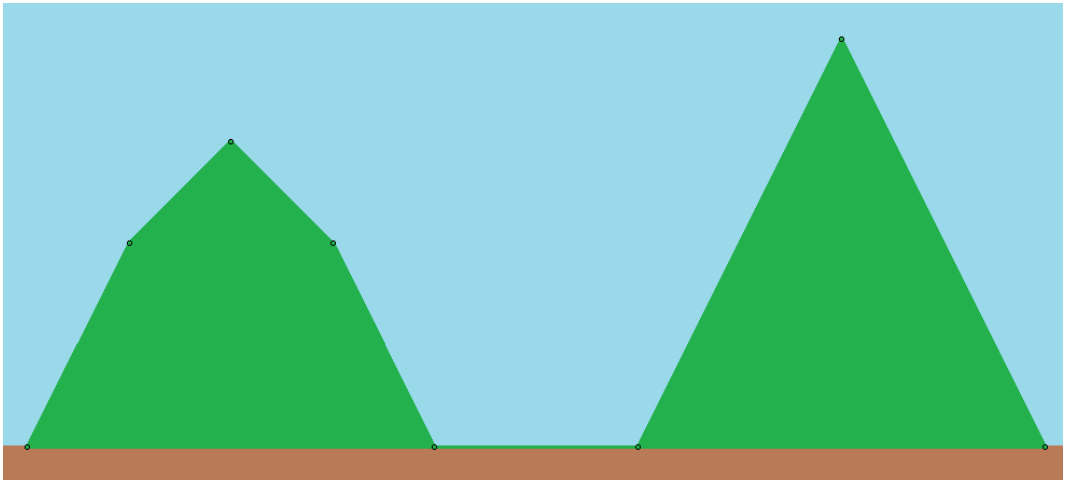
Output

각 테스트 케이스마다 까마귀가 이동한 총 이동거리를 출력한다. 절대/상대 오차가 10^{-9} 이하인 경우 정답으로 인정된다.

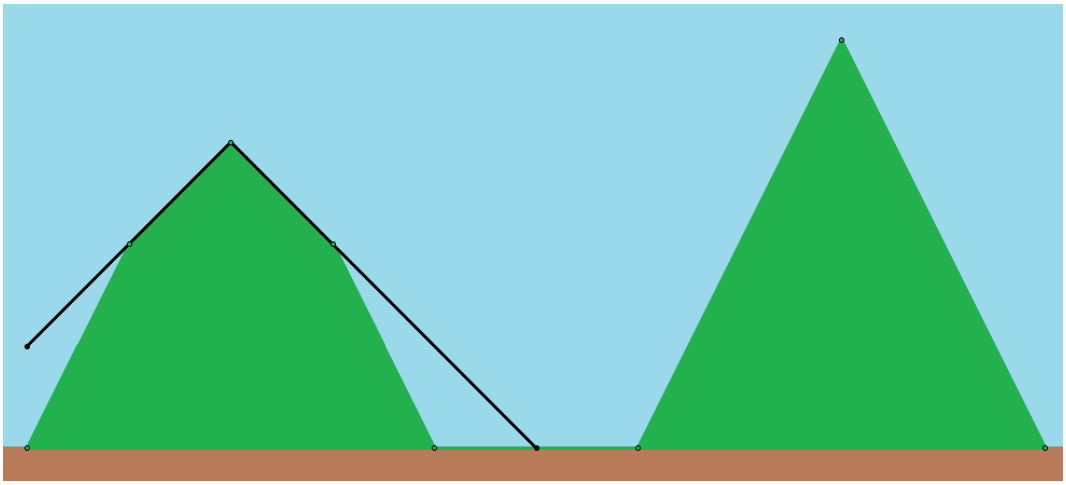
Sample input and output

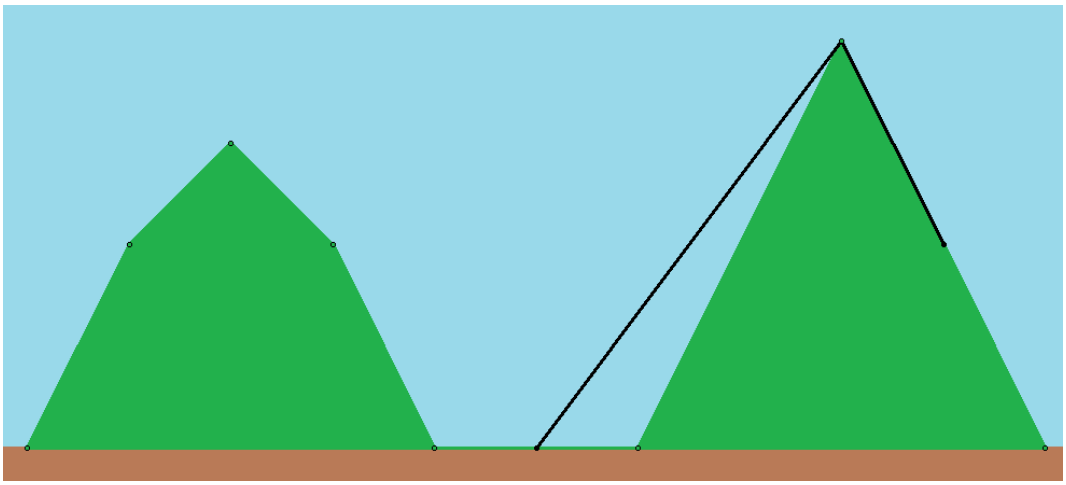
standard input	standard output
1 8 0 0 1 2 2 3 3 2 4 0 6 0 8 4 10 0 3 0 1 5 0 9 2	14.307135789365

Notes



예제 입력은 위의 그림과 같다. 초록색 부분은 산이고 갈색 부분은 땅이다. 까마귀는 하늘색 부분이나 산과 땅의 변을 통해 이동할 수 있다. 아래 그림은 까마귀의 두 이동을 나타낸다. 첫 번째 이동의 이동 거리는 $5\sqrt{2}$, 두 번째 이동의 이동 거리는 $5+\sqrt{5}$로 이를 모두 더하면 약 14.307135789365이다.





Problem K. Forest (외부 기출)

The government is planing to build a walkway for tourists in the middle of an oak forest. The forest can be represented as plane with N special lattice points representing oaks.

The walkway is represented as a rectangle with sides parallel to the axes. If the sides of walkway rectangle intersect any oak lattice points, such oaks need to be axed down. Oaks inside the rectangle do not represent problems and need not be cut down.

Bryan is the state secretary of Forestry and an passionate nature lover, so he ordered the secretary of Tourism to provide him with a list of P possible rectangle walkways that are attractive enough to draw in tourists.

Bryan plans to select the walkway that needs the smallest amount of oak trees to be cut down. Since we also like trees, would you be so kind and write a program that will determine the number of oaks that will be cut down for each walkway. Remember only the oaks intersecting the sides of the rectangle need to be cut.

Input

The first line of input contains one integer N ($1 \leq N \leq 300,000$), number of oaks.

The next N lines contain two integers X and Y ($1 \leq X, Y \leq 10^9$) representing coordinates of oaks. There will be at most one oak on each lattice point.

The next line contains one integer P ($1 \leq P \leq 100,000$), number of walkways.

The next P lines contain four integers X_1, Y_1, X_2 and Y_2($1 \leq X_1 < X_2 \leq 10^9, 1 \leq Y_1 < Y_2 \leq 10^9$) representing coordinates of the lower left (X_1, Y_1) and upper right (X_2, Y_2) corner of the rectangle.

Output

Output P integers, one per line, the number of oaks that need to be cut down for each walkway in the order they are presented in the input.

Sample input and output

standard input	standard output
6 1 2 3 2 2 3 2 5 4 4 6 3 4 2 2 4 4 2 2 6 5 3 3 5 6 5 1 6 6	3 4 0 1

Notes

출처: COCI 2010 1번 HRASTOVI