

2017
서울대학교 프로그래밍 경시대회
Division 1

주최 및 주관



후원



서울대학교 컴퓨터공학부
Seoul National University
Dept. of Computer Science and Engineering

2017년 9월 10일

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.
 - C reference: <http://en.cppreference.com/w/c>
 - C++ reference: <http://en.cppreference.com/w/cpp>
 - Java documentation: <http://docs.oracle.com/javase/8/docs/api/>
 - Python3 reference: <https://docs.python.org/3/reference/>
- 대회는 DOMjudge(<https://www.domjudge.org/>)를 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 제출하실 답안 코드는 C11, C++11, C++14, Java, Python 3 로만 작성하셔야 합니다. 단, 출제자는 C++14를 제외한 다른 언어로 주어진 시간 제한과 메모리 제한을 지키며 올바른 답을 내는 코드를 작성할 수 있다는 보장을 하지 않습니다.
- 모든 입력은 표준 입력으로 주어지며, 모든 출력은 표준 출력으로 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 출력에 주의하십시오. 프로세스가 0 이 아닌 리턴 코드를 되돌리는 경우나 표준 오류 스트림에 출력을 하는 경우 “RUN-ERROR” 를 받게 됩니다.
- 문제에 대한 질의 사항은 대회 페이지의 질문 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 “답변을 드릴 수 없습니다” 로 대답될 수 있으므로 유의하십시오.

채점 결과에 대하여

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

COMPILER-ERROR 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

RUN-ERROR 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

TIMELIMIT 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

WRONG-ANSWER 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

만약 여러 가지의 원인으로 인해 “CORRECT” 가 아닌 다른 결과를 얻으셨다면, 그 중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 비정상적인 동작도 수행하는 코드를 제출하신 경우 대부분 “RUN-ERROR” 를 받으시게 되지만, 경우에 따라서 “WRONG-ANSWER” 를 받을 수도 있습니다.

문제별 시간 제한은 다음과 같습니다.

- A : 1.0 s
- B : 1.0 s
- C : 2.0 s
- D : 1.5 s
- E : 4.0 s
- F : 1.0 s
- G : 1.0 s
- H : 1.0 s
- I : 1.0 s
- J : 2.0 s
- K : 1.0 s

문제별 메모리 제한은 모든 문제에 대해 512MB로 동일합니다.

Problem A. 전생했더니 슬라임 연구자였던 건에 대하여 (Hard)

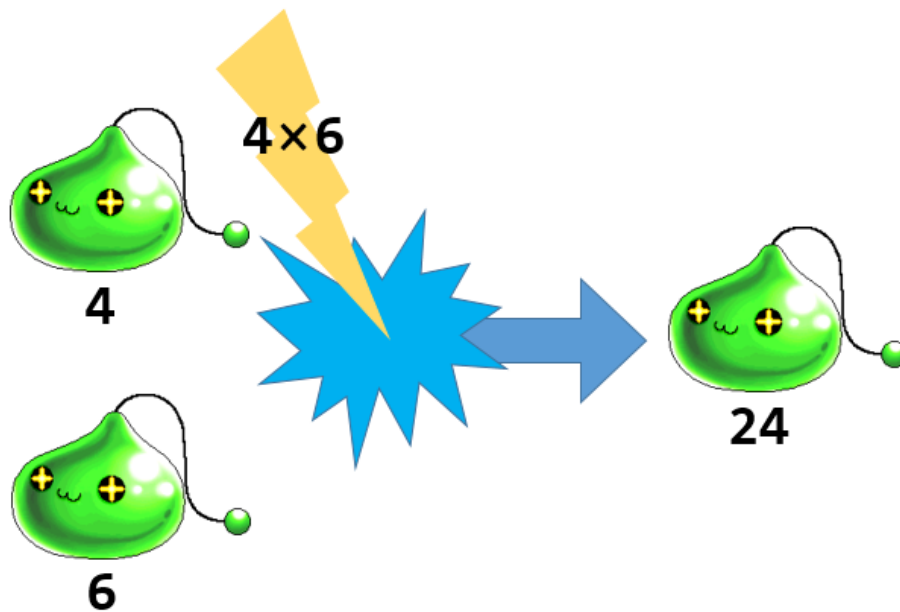
안녕? 내 이름은 ntopia!

나는 원래 지구에 살고 있던 평범한 20대 청년이었어. 어느 날 길을 걷다가 괴한의 칼에 찔려 죽어버렸어. 그런데 이게 무슨 일이라! 정신을 차려보니 이세계에 떨어져 버렸지 뭐야. 여기에서 나는 슬라임을 전문으로 연구하는 슬라임 연구자가 되어버린 것 같아. 나는 지금 아주 중요한 연구를 진행하고 있어. 이 연구가 성공하면 나는 내가 살던 세계로 돌아갈 수 있게 될 거야. 이 연구를 도와주지 않겠니?

이곳의 슬라임은 모두 슬라임 에너지라는 것을 갖고 있고 그 양은 2 이상의 자연수로 표현돼. 나는 슬라임을 합성했을 때 슬라임 에너지가 어떻게 변화하는지에 대해 연구하고 있어.

슬라임 합성 과정은 2마리를 합성해서 1마리를 만들어내는 식으로 이루어져. A 만큼의 슬라임 에너지를 가진 슬라임과 B 만큼의 슬라임 에너지를 갖고 있는 슬라임이 있었다고 해보자. 이 슬라임 2마리를 합성하면 슬라임 에너지가 $A \times B$ 인 슬라임을 만들 수 있어.

그리고 슬라임 합성 기술이 아직 완벽하지 않아서 슬라임을 합성할 때마다 크나큰 전기 에너지가 필요해. 구체적으로, A 만큼의 슬라임 에너지를 가진 슬라임과 B 만큼의 슬라임 에너지를 가진 슬라임을 합성하려면 $A \times B$ 만큼의 전기 에너지가 필요해.



에너지가 4인 슬라임과 에너지가 6인 슬라임을 합성한 모습. 4×6 의 전기 에너지를 사용해 슬라임 에너지가 24인 슬라임이 합성되었다.

나에겐 지금 N 마리의 슬라임이 있어. 이 슬라임들을 모두 적절히 합성해서 1마리의 슬라임으로 만들려고 해. 그런데 내가 소속된 연구소에서 각 합성 단계마다 필요한 전기 에너지들을 모두 곱한 값을 나에게 비용으로 청구하겠다고 했지 뭐야. 그래서 이 값이 최소가 되도록 합성을 적절히 수행하는 것이 내 연구의 목표야.

내 연구를 도와줘! 부탁이야!!

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어지고, 이어서 T 개의 테스트 케이스가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 슬라임의 수 N ($1 \leq N \leq 60$)이 주어지고, 두 번째 줄에는 N 개의 자연수가 주어진다. i 번째 자연수 C_i ($2 \leq C_i \leq 2 \times 10^{18}$)는 i 번째 슬라임의 슬라임 에너지를 나타낸다. 끝까지 합성하고 난 후에 생기는 슬라임의 에너지의 양이 2×10^{18} 이하라는 것이 보장된다.

모든 테스트 케이스에 대한 N 의 총합이 1,000,000을 넘지 않음이 보장된다.

Output

각 테스트 케이스마다 슬라임을 끝까지 합성했을 때 청구될 비용의 최솟값을 1,000,000,007로 나눈 나머지를 출력한다. 전기 에너지가 전혀 필요하지 않은 경우엔 1을 출력한다.

Sample input and output

standard input	standard output
2	270950400
5	1
3 10 2 8 14	
1	
13	

Problem B. 관악산 등산

서울대학교에는 “누가 조국의 미래를 묻거든 고개를 들어 관악을 보게 하라”라는 유명한 문구가 있다. 어느 날 Unused는 Corea에게 조국의 미래를 물었고, Corea는 직접 관악산에 올라가 조국의 미래를 보고 답해 주기로 했다.

관악산의 등산로는 1부터 N 까지의 서로 다른 번호가 붙어 있는 N 개의 쉼터와 두 쉼터 사이를 오갈 수 있는 M 개의 길들로 이루어져 있다. Corea는 지면에서부터 산을 오르는 것은 너무 귀찮다고 생각했기 때문에, 케이블카를 타고 임의의 쉼터에서 내린 다음 등산을 시작하기로 했다. Corea는 항상 더 높은 곳을 지향하기 때문에, 쉼터에 도착하면 그 쉼터와 직접 연결된 더 높은 쉼터로 향하는 길들 중 하나를 골라서 그 길을 따라 이동한다. 만약 그런 길이 없다면 등산을 마친다.

관악산의 쉼터들에는 조국의 미래를 볼 수 있는 전망대가 하나씩 설치되어 있다. Corea는 최대한 많은 쉼터를 방문해서 조국의 미래를 많이 보고 Unused에게 전해 주기로 했다. 관악산의 지도가 주어질 때, Corea가 각각의 쉼터에서 출발해서 산을 오를 때 최대 몇 개의 쉼터를 방문할 수 있는지 구하여라.

Input

첫 번째 줄에 등산로에 있는 쉼터의 수 N ($2 \leq N \leq 5,000$)과 두 쉼터 사이를 연결하는 길의 수 M ($1 \leq M \leq 100,000$)이 주어진다.

두 번째 줄에는 각 쉼터의 높이를 나타내는 N 개의 정수가 번호 순서대로 주어진다. 각 쉼터의 높이는 1 이상 1,000,000 이하이며 서로 다르다.

세 번째 줄부터 M 개의 줄에 걸쳐 각각의 길이 연결하는 두 쉼터의 번호가 공백으로 구분되어 주어진다. 쉼터의 번호는 1 이상 N 이하의 정수이다. 양 끝점이 같은 쉼터인 길은 없으며, 임의의 두 쉼터를 연결하는 길이 여러 개 존재할 수 있다.

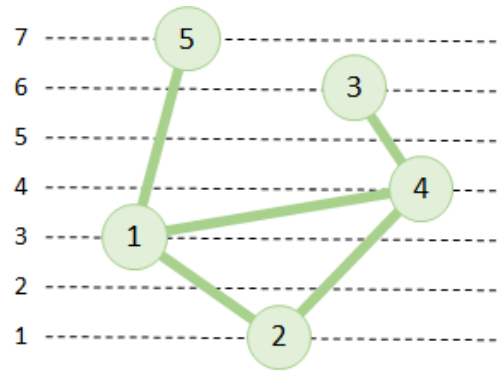
Output

N 개의 줄에 걸쳐 n 번째 줄에 Corea가 n 번 쉼터에서 출발해서 산을 오를 때 최대 방문할 수 있는 쉼터의 개수를 출력한다.

Sample input and output

standard input	standard output
5 5	3
3 1 6 4 7	4
1 4	1
2 1	2
3 4	1
4 2	
5 1	

Notes

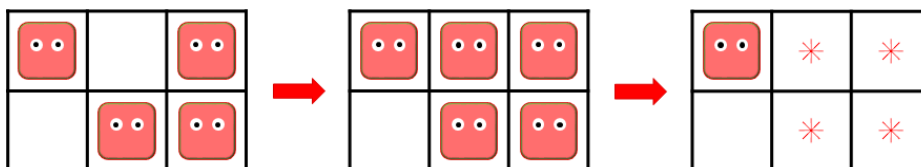


2번 쉼터에서 출발하면 1번, 4번, 3번 쉼터를 차례대로 방문할 때 가장 많은 쉼터를 방문할 수 있다.

5번 쉼터는 3번 쉼터보다 높은 곳에 있지만 길 하나로 연결되어 있지 않으므로 3번 쉼터에서 5번 쉼터로 이동할 수 없다.

Problem C. 넴모넴모 (Hard)

넴모는 $\text{뿌} \times \text{뿌} \times \text{뿌}$ 게임에 깊은 감명을 받아, 직사각형 모양의 격자판과 “넴모”라는 수수께끼의 생물을 이용하는 “넴모넴모”라는 게임을 만들었다. 이 게임의 규칙은 아주 간단하다. 격자판의 비어 있는 칸을 임의로 골라 “넴모”를 하나 올려놓거나, “넴모”가 올라간 칸 네 개가 2×2 사각형을 이루는 부분을 찾아 그 위에 있는 “넴모”들을 모두 없애는 것을 질릴 때까지 반복하면 된다.



하지만 안타깝게도 게임은 정말 재미가 없었고, 넴모는 아주 빨리 질려 버리고 말았다. 실망한 넴모는 게임을 적당히 플레이하다가, “넴모”를 없애고 싶는데 격자판 위에 없앨 수 있는 “넴모”가 없으면 게임을 그만두기로 했다. 넴모가 게임을 그만두었을 때 나올 수 있는 “넴모”의 배치의 가짓수를 구하여라.

Input

첫 번째 줄에 격자판의 행의 개수 N , 열의 개수 M ($1 \leq N, M \leq 300, 1 \leq N \times M \leq 300$)이 공백으로 구분되어 주어진다.

Output

첫 번째 줄에 주어진 격자판에서 나올 수 있는, “넴모”들이 올라간 칸이 2×2 사각형을 이루지 않는 모든 배치의 가짓수를 1,000,000,007로 나눈 나머지를 출력한다.

Sample input and output

standard input	standard output
2 2	15
3 5	22077
5 7	185495795

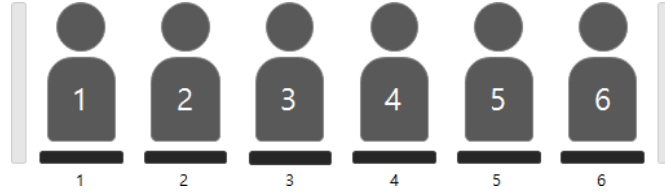
Notes

2×2 격자판에 2×2 사각형을 이루지 않도록 “넴모”들을 배치하는 방법은 모든 경우($2^4 = 16$) 중 네 칸 모두에 “넴모”가 올라가 있는 경우를 제외한 15가지가 있다.

5×7 격자판에 2×2 사각형을 이루지 않도록 “넴모”들을 배치하는 방법은 총 11,185,495,872가지가 있다.

Problem D. 셔틀버스

서울대학교 내부를 운행하는 셔틀버스에는 한쪽 벽면에 N 개의 좌석이 일렬로 놓여 있다. 각 좌석은 가장 왼쪽 좌석부터 시작하여 1번부터 N 번까지의 번호가 붙어 있다. 이 버스는 학교 입구에서 N 명의 학생들을 태운 뒤 출발하고, 학생들은 각자 하나의 좌석을 골라 앉는다. 편의상 출발할 때 i 번 좌석에 앉은 학생을 i 번 학생으로 부르기로 한다.



학생들은 칸막이에 기대서 조는 것을 좋아하기 때문에 되도록 양쪽 끝자리에 앉고 싶어 한다. 그래서 바로 옆에 앉아 있던 학생이 내리거나 다른 자리로 옮겨 가서 좌석이 비었을 때 그 좌석으로 옮겨 앉는 게 양쪽 끝자리에 더 가까워질 경우, 즉 1번 좌석과 N 번 좌석 중 더 가까운 좌석까지의 거리가 줄어들 경우 그 좌석으로 옮겨 앉는다. 한 학생이 버스에서 내리는 즉시 모든 학생이 이 규칙에 따라 이동한다.

셔틀버스 기사 찬수는 버스에서 내리는 학생들을 보면서 지금 어떤 좌석에 어떤 학생이 앉아 있는지 궁금해졌다. 찬수를 위해 아래의 두 가지 연산을 입력되는 순서대로 수행하는 프로그램을 작성해 주자.

1. $1 \ x$: x 번 학생이 버스에서 내린다. 이 학생은 버스에 타고 있던 학생임이 보장된다.
2. $2 \ x$: x 번 좌석에 앉아 있는 학생의 번호를 출력한다. 좌석이 비어 있을 경우는 0을 출력한다.

찬수가 운전하는 버스는 차고지로 들어가는 버스이기 때문에 새로운 학생을 태우지는 않는다.

Input

첫 번째 줄에 셔틀버스에 있는 좌석의 수 N ($1 \leq N \leq 200,000$), 처리해야 하는 연산의 수 M ($1 \leq M \leq 400,000$)이 주어진다.

두 번째 줄부터 M 개의 줄에 걸쳐 각 쿼리의 종류(1 또는 2)와 x ($1 \leq x \leq N$) 값이 공백을 사이에 두고 주어진다. 2번 쿼리가 하나 이상 존재함이 보장된다.

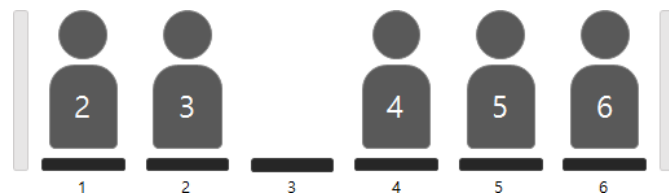
Output

각 2번 쿼리의 결과를 입력되는 순서대로 한 줄에 하나씩 출력한다.

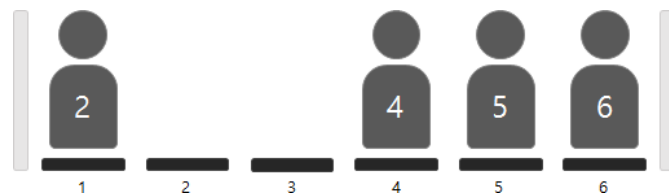
Sample input and output

standard input	standard output
6 9	2
2 2	2
1 1	6
2 1	4
2 6	4
1 3	0
2 4	
1 5	
2 5	
2 3	

Notes



1번 학생이 내리면 2번 학생과 3번 학생이 순서대로 이동한다. 4번 학생은 3번 좌석으로 옮겨 앉더라도 양 끝 좌석까지의 거리가 변하지 않기 때문에 이동하지 않는다.



3번 학생이 내리면 4번 학생은 2번 좌석으로 이동하는 것이 이득이지만, 바로 옆자리가 아니므로 이동하지 않는다.



5번 학생이 내리면 4번 학생이 이동한다.

Problem E. 데굴데굴

종훈이는 질 좋은 문제와 푸짐한 상품이 주어지는 서울대학교 프로그래밍 경시대회에 출전했다. 하지만 참가 신청을 할 때 실수를 하는 바람에 숨겨진 난이도인 Div. 0 난이도에 신청하고 말았고, $N = 1,000$ 짜리 Travelling salesman problem을 풀면서 고통받고 있었다. 결국, 그 문제를 NP(Not my Problem)-hard 문제로 결론지은 종훈이는 대회장에 가지고 온 물병을 가지고 놀면서 남은 시간을 보내기로 했다.

종훈이가 가지고 온 물병은 밑면이 볼록다각형인 직각기둥 모양이고, 물이 조금 들어 있다. 종훈이는 물병을 완전히, 즉 두 밑면에 속하지 않은 변이 모두 밑면과 평행하도록 눕혀 놓고 데굴데굴 굴려 보았다. 그리고 물병이 구르면서 물병의 밑면에서 물이 차지하는 영역이 다양한 모양으로 바뀌는 걸 지켜보다가, 문득 이 영역을 자기가 원하는 개수만큼의 변을 가진 다각형으로 만들 수 있는지 궁금해졌다. 종훈이는 자신이 이 문제를 풀 수 있을 거라 생각했지만, 이미 너무 지쳐 있었기 때문에 대신 Div. 1 대회장에서 열심히 문제를 풀고 있는 당신에게 도움을 요청했다.

종훈이는 수시로 물을 마시거나 정수기에서 다시 물을 떠 오고, 그 때마다 당신에게 도움을 요청할 것이다. 편의를 위해 물병의 높이는 1cm 이며, 물병의 물은 물병을 기울이는 즉시 평형을 되찾는다고 가정한다.

Input

첫 번째 줄에 물병의 밑면을 이루는 꼭짓점의 수 N ($3 \leq N \leq 500,000$), 종훈이가 도움을 요청한 횟수 M ($1 \leq M \leq 50$)이 주어진다.

두 번째 줄부터 N 개의 줄에 걸쳐 물병의 밑면을 이루는 꼭짓점의 x 좌표와 y 좌표가 공백으로 구분되어 주어진다. 이는 물병의 밑면을 1cm 단위로 격자가 그려진 좌표평면에 본뜬 뒤 임의의 꼭짓점에서 시작하여 반시계방향으로 돌면서 만나는 꼭짓점들의 좌표를 순서대로 기록한 값이다. 각 좌표값은 모두 정수이며 $-100,000,000$ 이상 $100,000,000$ 이하이다. 세 꼭짓점이 한 직선 위에 있는 경우는 없다.

$N+2$ 번째 줄부터 M 개의 줄에 걸쳐 각각의 요청에 대해 물병에 들어 있는 물의 양을 의미하는 정수 W ($1 \leq W \leq$ 물병의 부피, 단위는 cm^3)가 주어진다.

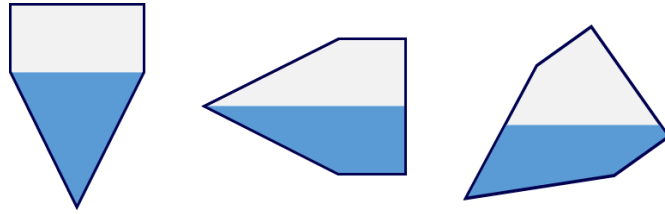
Output

M 개의 줄에 걸쳐 m 번째 줄에 m 번째 요청에 대해 물병을 눕혀 놓고 굴렸을 때 물병의 밑면에서 물이 차지하는 영역이 가질 수 있는 가장 적은 변의 수와 가장 많은 변의 수를 공백으로 구분하여 출력한다.

Sample input and output

standard input	standard output
5 2	3 5
-5 2	4 6
1 -1	
4 -1	
4 5	
1 5	
18	
24	

Notes



위의 그림은 첫 번째 요청에 대해 물이 차지하는 영역이 삼각형, 사각형, 오각형을 이루는 경우이다.

Problem F. 전자기기

종범이의 집에는 N 개의 전자기기들이 있다. i 번 전자기기는 X_i 개의 장치로 구성되어 있고, i 번 전자기기의 j 번 장치는 $A_{i,j}$ 의 전력을 공급받아야 정상적으로 동작한다. $A_{i,j}$ 미만의 전력을 공급받으면 전력이 부족해서 동작하지 않고, $A_{i,j}$ 초과 전력의 전력을 공급받으면 장치가 망가진다. i 번 전자기기의 장치들 중 Y_i 개 이상의 장치가 정상적으로 동작하면 i 번 전자기기를 사용할 수 있다. 종범이는 장치들 중 하나에 전력을 공급할 수 있는 M 개의 전원 공급 장치가 있다. k 번 전원 공급 장치는 Z_k 가지 서로 다른 전력들 중 하나를 골라 정확히 하나의 장치에 공급할 수 있으며, k 번 전원 공급 장치가 공급할 수 있는 전력들 중 l 번째로 작은 것은 $B_{k,l}$ 이다.

종범이는 새 전자기기를 들일 때마다 전원 공급 장치를 아무거나 사서 연결해 N 개의 전자기기를 모두 잘 사용하고 있었다. 종범이는 이사를 할 일이 생겼고, 가지고 있는 전자기기들과 전원 공급 장치를 모두 들고 이사를 했다. 그런데 깜빡하고 어느 전자기기의 어느 장치에 어떤 전원 공급 장치를 연결했는지 기록하지 않고 이사를 해 버려 전자기기를 사용할 수 없게 되었다. 이사한 기념으로 리X오X레XX와 스타XXXX를 하려던 종범이는 전자기기를 사용할 수 없다는 사실이 너무 슬펐다. 종범이를 도와 어느 전자기기의 어느 장치에 어떤 전원 공급 장치를 연결하면 모든 전자기기를 사용할 수 있는지를 찾아주자.

Input

1행에 N 이 주어진다. ($1 \leq N \leq 100$)

2행부터 $2N + 1$ 행까지 N 개의 전자기기에 대한 설명이 주어진다.

- $2i$ 행에는 X_i 와 Y_i 가 주어진다. ($1 \leq Y_i \leq X_i \leq 100$)
- $2i + 1$ 행에는 X_i 개의 정수 $A_{i,j}$ 가 주어진다. ($1 \leq A_{i,j} \leq 10^9$)

$2N + 2$ 행에 M 이 주어진다. ($1 \leq M \leq 10,000$)

$2N + 3$ 행부터 $2N + 2M + 2$ 행까지 M 개의 전원 공급 장치에 대한 설명이 주어진다.

- $2N + 2k + 1$ 행에는 Z_k 가 주어진다. ($1 \leq Z_k \leq 10$)
- $2N + 2k + 2$ 행에는 Z_k 개의 정수 $B_{k,l}$ 이 주어진다. ($1 \leq B_{k,l} \leq 10^9$)

모든 전자기기를 사용할 수 있는 연결 방법이 하나 이상 존재함이 보장된다.

Output

1행에 서로 연결해야 하는 전자기기의 장치와 전원 공급 장치의 연결 개수 W 를 출력한다.

2행부터 $W + 1$ 행까지 각 행에 (i, j, k, l) 순서쌍을 출력한다. 이는 i 번 전자기기의 j 번 장치와 k 번 전원 공급 장치를 연결하고, k 번 전원 공급 장치가 공급할 수 있는 전력들 중 l 번째로 작은 것을 선택해야 한다는 것을 의미한다.

한 전자기기의 장치가 여러 번 등장하거나 한 전원 공급 장치가 여러 번 등장하면 안 된다. 또한 각 연결에 대해서 전자기기의 장치가 요구하는 전력과 전원 공급 장치가 공급하는 전력의 양이 동일해야 한다.

Sample input and output

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

Notes

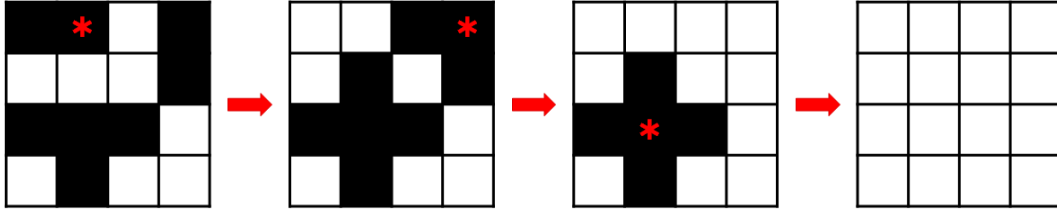
총 5개의 연결이 있다.

1번 전자기기의 2번 장치에는 5번 전원 공급 장치로 1의 전력을 공급했고, 1번 전자기기의 3번 장치에는 1번 전원 공급 장치로 3의 전력을 공급했다. 1번 전자기기의 장치들 중 2개 이상이 동작하므로 1번 전자기기를 사용할 수 있다.

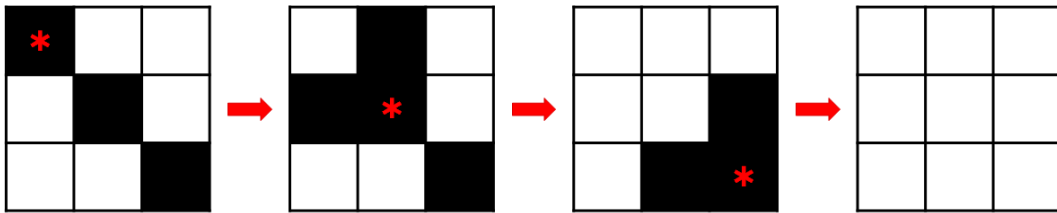
2번 전자기기의 1번 장치에는 2번 전원 공급 장치로 2의 전력을 공급했고, 2번 전자기기의 2번 장치에는 4번 전원 공급 장치로 4의 전력을 공급했고, 2번 전자기기의 4번 장치에는 3번 전원 공급 장치로 4의 전력을 공급했다. 2번 전자기기의 장치들 중 3개 이상이 동작하므로 2번 전자기기를 사용할 수 있다.

Problem G. 타일 뒤집기 (Hard)

지구이는 신기한 게임판을 가지고 있다. 이 게임판에는 한 면은 검은색, 한 면은 흰색으로 칠해진 타일이 N 행 N 열로 배치되어 있다. 각 타일은 제자리에서 뒤집을 수 있는데, 타일 하나를 뒤집으면 그 타일과 상하좌우로 인접한 타일들이 같이 뒤집힌다. 지구이는 타일들이 무작위로 배치된 게임판에서 타일들을 적당히 뒤집어서 모든 타일이 흰색 면이 위를 향하도록 만드는 놀이를 좋아한다.



어느 날, 지구이가 게임판을 가지고 놀다가 자리를 비운 사이 지구이의 동생이 이 게임판을 발견했다. 지구이의 동생은 놀이의 규칙을 모르기 때문에, 그냥 처음 상태에서 검은색 면이 위를 향하고 있는 타일들을 전부 한 번씩 뒤집어 보았다. 그러자 놀랍게도 모든 타일이 흰색 면이 위를 향하게 되었다!



처음 상태

돌아온 지구이는 동생에게 놀이의 규칙을 알려 주려고 했지만, 그 전에 동생이 즐거워하는 모습을 더 보고 싶어져서 같은 특징을 갖는 게임판을 몇 번 더 만들어 주기로 했다. 지구이는 멋진 해커이기 때문에 게임판의 규칙을 따르지 않고 원하는 타일들만 따로 뒤집어서 원하는 색 배치를 만들 수 있다. 하지만 아무 조건 없이 타일을 뒤집는 것은 별로 재미가 없었기 때문에, 지구이는 게임판에서 몇 개의 타일들은 뒤집지 않고 원하는 배치를 만들어 보기로 했다.

Input

첫 번째 줄에 게임판의 크기 N ($1 \leq N \leq 1,000$)이 주어진다.

두 번째 줄부터 N 개의 줄에 걸쳐 게임판의 타일들의 상태를 나타내는 N 글자의 문자열이 주어진다. 문자열은 #와 ., -만 이루어져 있으며, #는 검은색 면이 위를 향하도록 고정된 타일, .는 흰색 면이 위를 향하도록 고정된 타일, -는 지구이가 마음대로 뒤집을 수 있는 타일을 의미한다.

Output

N 개의 줄에 걸쳐 지구이의 동생이 검은색 면이 위를 향하고 있는 타일들을 전부 한 번씩 뒤집어서 모든 타일이 흰색 면이 위를 향하도록 만들 수 있는 게임판의 모양을 출력한다. 입력 조건과 마찬가지로 검은색 면이 위를 향하고 있는 타일은 #, 흰색 면이 위를 향하고 있는 타일은 .로 나타낸다.

답이 여러 가지일 경우 아무것이나 출력하고, 답이 존재하지 않을 경우 `thinking_face`를 출력한다.

Sample input and output

standard input	standard output
4 -.- ---# --.- .---	#.#. ...# #... .##
7 ---.- -----.- --#----- .----- -----# -.- ----#--	#.#.####.# #.#.### ..#.#.. #####.# #.#.... ###.##
5 ----# #--.- -#--- ----. --.-	thinking_face

Problem H. 홍삼 게임 (Hard)

은하는 술과 게임과 술 게임을 좋아한다. 그중에서도 가장 좋아하는 술 게임은 “홍삼 게임”이다. 이 게임은 N 명의 사람이 테이블에 원형으로 둘러앉은 상태에서 하는 게임이고, 규칙은 다음과 같다.

1. 은하가 서로 다른 두 사람을 지목한다.
2. 지목당한 두 사람이 동시에 테이블에 앉은 사람들 중 한 사람씩을 골라서 지목한다.
3. 만약 두 사람이 같은 사람을 지목했을 경우 게임이 끝난다. 그렇지 않을 경우 2번으로 돌아간다.

SNUPC가 끝난 뒤 참가자들은 근처 술집으로 뒤풀이를 하러 갔고, 은하의 주도 아래 홍삼 게임을 하게 되었다. 하지만 사람이 너무 많이 모이는 바람에 누가 누구를 지목하는지 잘 보이지 않아서 게임이 수시로 중단되었다. 이 상황을 보다 못한 은하의 친구 은서는 홍삼 게임의 규칙을 수정한 “질서 있는 홍삼 게임”을 제안하였다. 이 게임의 규칙은 다음과 같다.

1. 은하가 서로 다른 두 사람을 순서대로 지목한다. 먼저 지목당한 사람은 지목권 A, 두 번째로 지목당한 사람은 지목권 B를 갖는다.
2. 지목권 A를 가진 사람이 자신의 왼쪽 또는 오른쪽으로 정확히 D_A 만큼 떨어진 사람 한 명을 지목하여 자신의 지목권을 넘긴다.
3. 만약 지목당한 사람이 이미 지목권 B를 가지고 있었을 경우 게임이 끝난다.
4. 지목권 B를 가진 사람이 자신의 왼쪽 또는 오른쪽으로 정확히 D_B 만큼 떨어진 사람 한 명을 지목하여 자신의 지목권을 넘긴다.
5. 만약 지목당한 사람이 이미 지목권 A를 가지고 있었을 경우 게임이 끝난다. 그렇지 않을 경우 2번으로 돌아간다.

은서의 제안 덕분에 참가자들은 질서 있게 홍삼 게임을 즐길 수 있게 되었다. 하지만 은하가 몇 시간 내내 계속 홍삼 게임을 돌리자 참가자들은 지쳐 갔고, 은하가 누구를 지목하고 지목 간격을 어떻게 설정하든 최대한 게임을 빠르게 끝내려고 하게 되었다. 불쌍한 뒤풀이 참가자들을 홍삼 지옥에서 구해 주자.

편의를 위해 참가자들에게는 1번부터 N 번까지 반시계방향으로 번호가 붙어 있다고 가정한다. 즉 i 번 참가자의 바로 왼쪽에는 $i - 1$ 번, 바로 오른쪽에는 $i + 1$ 번 참가자가 앉아 있다. 예외로 1번 참가자의 바로 왼쪽에는 N 번 참가자가, 마찬가지로 N 번 참가자의 바로 오른쪽에는 1번 참가자가 앉아 있다.

Input

첫 번째 줄에 “질서 있는 홍삼 게임”의 참가자의 수 N ($2 \leq N \leq 500,000$), 은하가 먼저 지목한 사람의 번호 A 와 두 번째로 지목한 사람의 번호 B ($1 \leq A, B \leq N$, $A \neq B$), 각 지목권의 지목 간격을 나타내는 정수 D_A , D_B ($1 \leq D_A, D_B \leq N - 1$)이 공백을 사이에 두고 순서대로 주어진다.

Output

첫 번째 줄에 입력된 게임을 최대한 빠르게 끝내고자 할 때 필요한 최소 지목 횟수를 출력한다. 만약 끝낼 수 없는 게임일 경우 Evil Galaxy를 출력한다.

Sample input and output

standard input	standard output
6 5 1 1 2	3
4 1 2 2 2	Evil Galazy

Notes

첫 번째 예시는 다음과 같은 순서로 진행하면 3번의 지목으로 끝낼 수 있다.

1. 지목권 A를 가진 5번 참가자는 4번 또는 6번 참가자를 지목할 수 있다. 이 중 4번 참가자를 지목하여 지목권을 넘긴다.
2. 지목권 B를 가진 1번 참가자는 5번 또는 3번 참가자를 지목할 수 있다. 이 중 3번 참가자를 지목하여 지목권을 넘긴다.
3. 지목권 A를 가진 4번 참가자가 3번 참가자를 지목하여 지목권을 넘기고 게임이 끝난다.

Problem I. 구간 합 최대

승현이는 음이 아닌 정수 N 개로 구성된 배열을 가지고 놀고 있다. 이 배열에는 M 개의 특별한 조건이 있는데, 그것은 길이가 L_i 인 어떤 연속한 구간을 잡아도 합이 S_i 를 넘지 않는다는 것이다. 승현이는 특별한 조건을 모두 만족하는 배열을 모두 만들어 놓고, 1 이상 N 이하의 모든 정수 K 에 대해 각 배열에서 길이가 K 인 모든 연속한 구간마다 원소의 합을 구하고 그 중 가장 큰 것을 찾아보았다. 하지만 승현이는 계산에 자신이 없어서 자신이 잘 구했는지 확신하지 못하고 있다. 승현이가 올바르게 구했는지 알아보자.

Input

첫 번째 줄에 배열을 구성하는 정수의 개수 N ($1 \leq N \leq 200,000$)과 특별한 조건의 개수 M ($1 \leq M \leq 200$)이 주어진다.

두 번째 줄부터 M 개의 줄에 걸쳐 특별한 조건을 의미하는 두 정수 L_i 와 S_i 가 주어진다. ($1 \leq i \leq M, 1 \leq L_i \leq N, 1 \leq S_i \leq 10^9$)

Output

N 줄에 걸쳐 K 번째 줄에 특별한 조건을 모두 만족하는 배열의 길이가 K 인 연속한 구간들 중 합이 가장 큰 구간의 구간 합을 출력한다.

Sample input and output

standard input	standard output
5 2	5
2 5	5
3 7	7
	10
	12

Notes

배열이 $[1, 4, 1, 0, 5]$ 일 때, 길이가 1인 연속한 구간들 중 합이 5인 것이 존재하고, 길이가 2인 연속한 구간들 중 합이 5인 것이 존재하고, 길이가 4인 연속한 구간들 중 합이 10인 것이 존재한다.

배열이 $[3, 2, 2, 1, 4]$ 일 때, 길이가 3인 연속한 구간들 중 합이 7인 것이 존재하고, 길이가 5인 연속한 구간들 중 합이 12인 것이 존재한다.

Problem J. 그림 그리기

상수는 N 행 M 열의 격자에 L 가지 색으로 그림을 그린다. 1행에서 시작해서 N 행까지, 같은 행에서는 1열에서 시작해서 M 열까지 순서대로 칠하는데, 1번 색부터 시작해서 L 번 색까지 순서대로 사용한 뒤 다시 1번 색부터 사용한다. i 번째 행의 높이는 H_i 이고 j 번째 열의 너비는 W_j 이다. 상수는 물감을 낭비하고 싶지 않기 때문에, 그림을 그리기 전에 각 색이 얼마나 필요한지가 궁금해졌다. 상수를 위해 각 색으로 칠할 넓이가 얼마나 되는지 구해보자.

Input

첫 번째 줄에 N, M, L 가 주어진다. ($1 \leq N, M, L \leq 123,456$)

두 번째 줄에 각 행의 높이를 나타내는 N 개의 자연수 H_i 가 공백을 사이에 두고 주어진다. H_i 의 합은 10^9 을 넘지 않는다. ($1 \leq i \leq N$)

세 번째 줄에 각 열의 너비를 나타내는 M 개의 자연수 W_j 가 공백을 사이에 두고 주어진다. W_j 의 합은 10^9 을 넘지 않는다. ($1 \leq j \leq M$)

Output

L 줄에 걸쳐 k 번째 줄에 k 번 색으로 칠할 넓이를 출력한다.

Sample input and output

standard input	standard output
4 5 3	63
2 4 3 4	70
2 4 1 2 5	49

Notes

상수가 그릴 그림은 다음처럼 생겼다.



Problem K. 정육면체를 사랑하는 사람

지학이는 정육면체를 정말 사랑한다. 정육면체 없이는 하루도 살아갈 수 없다. 그래서 자기 집에 정육면체를 산더미처럼 쌓아놓고 매일 그 정육면체들을 바라보며 행복한 인생을 살고 있다.

지학이가 자꾸 집에 정육면체를 들여놓자 집안이 엉망이 되어버려 화가 난 석환이는 지학이에게 특단의 조치를 내렸다. 집에 늘어놓은 정육면체들을 당장 정리하지 않으면 집에서 내쫓겠다는 것이다!

지학이는 가로, 세로, 높이가 1cm인 K 개의 정육면체들을 직육면체의 박스 1개에 모두 담아 정리해야 한다. 게다가 석환이는 까다로운 성격이라 다음과 같은 조건을 요구했다: 정육면체들의 각 면이 박스의 각 면과 평행하도록 놓여야 하고, 정육면체들끼리는 서로 맞닿아 있을 수는 있지만 서로 겹치거나 박스의 테두리를 뚫고 나오도록 놓을 수는 없다. 정육면체들을 어떻게 쌓고 배열하나에 따라 필요한 직육면체 박스의 크기가 바뀌게 된다는 것을 알 수 있다.

마지막 한 줄기 자비심이 남아있는 석환이는 정육면체들을 정리하는데 필요한 직육면체 박스를 직접 사다 주겠다고 했다. 대신 석환이의 돈을 쓰는 것이기 때문에 박스의 가격을 최소한으로 줄이는 방법을 알아내야 한다. 만약 가격이 최소가 아니라면 석환이는 박스를 사러 가지 않을 것이고 결국 지학이는 집에서 쫓겨나게 된다. 박스의 가격은 직육면체의 겹넓이에 비례하므로 결국 겹넓이가 최소가 되는 박스의 크기를 석환이에게 알려주어야 한다.

지학이가 집에서 쫓겨나게 되지 않도록 필요한 직육면체 박스의 겹넓이가 최소가 되는 크기를 구해 지학이에게 알려주자.

Input

첫 번째 줄에 자연수 K 개가 주어지는데, 이는 정육면체의 개수 K ($1 \leq K \leq 10^{18}$)를 나타낸다.

Output

첫 번째 줄에 가격이 최소가 되는 직육면체 박스의 가로, 세로, 높이를 cm 단위로 출력한다. 가격이 최소가 되는 직육면체 박스가 여러 개라면 (가로, 세로, 높이)로 순서쌍을 만들었을 때 사전 순으로 제일 앞서는 것을 출력한다.

Sample input and output

standard input	standard output
3	1 1 3
7	2 2 2