

2017
서울대학교 프로그래밍 경시대회
Division 1

주최 및 주관



후원



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.
- 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 0이 아닌 리턴 코드를 되돌리는 경우나 표준 오류 스트림에 출력을 하는 경우 “런타임 에러”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.

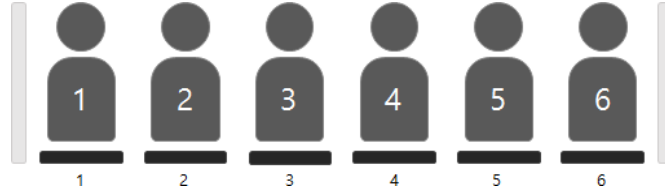
TIMELIMIT 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

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.

Problem A. 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 끝 자리에 앉고 싶어한다. 그래서 바로 옆에 앉아 있던 학생이 내리거나 다른 자리로 옮겨 가서 좌석이 비었을 때 그 좌석으로 옮겨 앉는 게 양쪽 끝 자리에 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 ($1 \leq N \leq 100,000$), 처리해야 하는 연산의 수 M ($1 \leq M \leq 200,000$)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.

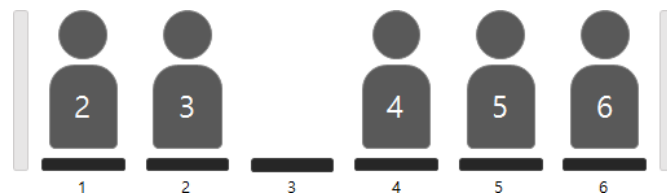
Output

각 2번 쿼리의 결과를 입력되는 순서대로 한 줄에 하나씩 출력한다.

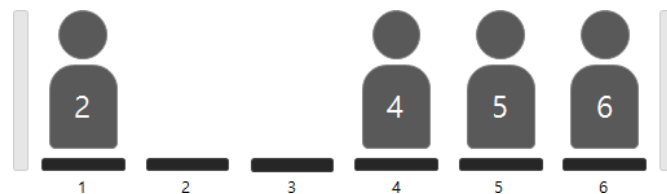
Sample input and output

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	

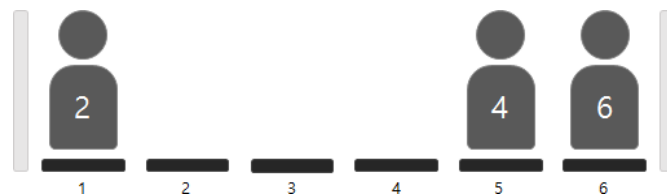
Notes



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 옆 자리가 아니므로 이동하지 않는다.



5번 학생이 내리면 4번 학생이 이동한다.

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 오고갈 수 있는 M 개의 길들로 이루어져 있다. Corea는 지면에서부터 산을 올라가는 것은 너무 귀찮다고 생각했기 때문에, 케이블카를 타고 임의의 쉼터에서 내린 다음 등산을 시작하기로 했다. 심지어 등산로 지도를 보는 것도 귀찮았던 Corea는 지금 있는 쉼터와 길 하나로 연결되어 있으면서 지금보다 더 높은 곳에 있는 쉼터를 하나 골라서 올라가는 것을 반복하면 산의 정상에 도착할 수 있을 거라고 생각했다. 물론 실제로는 그렇지 않을 수도 있다.

Corea가 임의의 쉼터에서 출발하여 자신의 생각에 따라 산을 오를 때, 최대 몇 개의 쉼터를 방문할 수 있는지 구하여라.

Input

첫 번째 줄에 등산로에 있는 쉼터의 수 N ($1 \leq N \leq 3000$)과 두 쉼터 사이를 연결하는 길의 수 M ($1 \leq M \leq 50,000$)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.

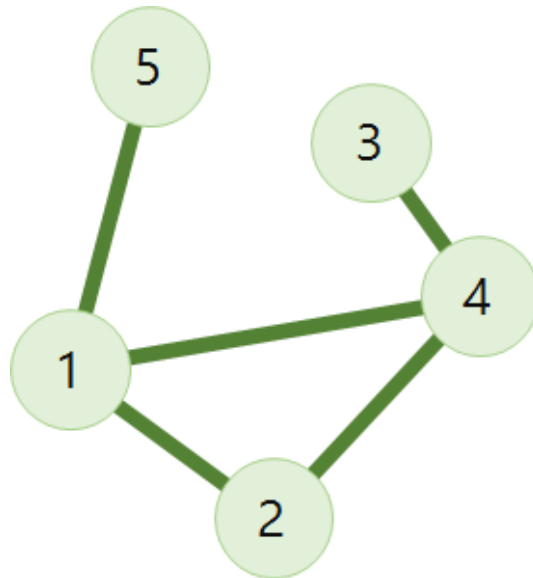
Output

Corea가 각 쉼터에서 출발할 때 최대로 방문할 수 있는 쉼터의 개수를 번호 순서대로 한 줄에 하나씩 N 줄에 걸쳐 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

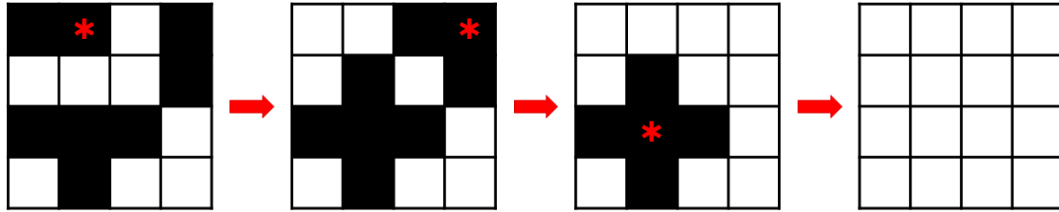


2번 쉼터에서 출발하면 1번, 4번, 3번 쉼터를 차례대로 방문할 때 가장 많은 쉼터를 방문할 수 있다.

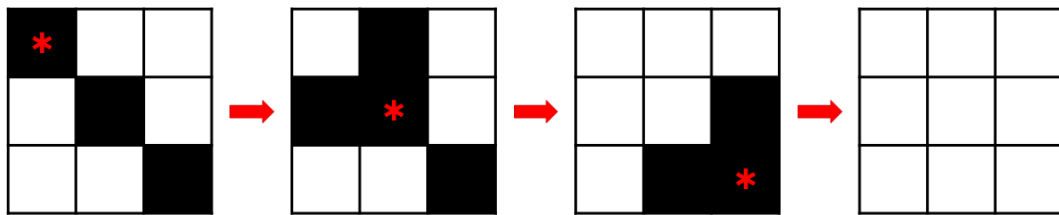
5번 쉼터는 3번 쉼터보다 높은 곳에 있지만 길 하나로 연결되어 있지 않으므로 3번 쉼터에서 5번 쉼터로 이동할 수 없다.

Problem C. 타일 뒤집기 2

석환이는 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. 석환이는 타일들이 무작위로 배치된 게임판에서 타일들을 적당히 뒤집어서 모든 타일이 흰색 면이 위를 향하도록 만드는 놀이를 좋아한다.



어느 날, 석환이가 게임판을 가지고 놀다가 자리를 비운 사이 석환이의 동생이 이 게임판을 발견했다. 석환이의 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!



돌아온 석환이는 동생에게 놀이의 규칙을 알려 주려고 했지만, 그 전에 동생이 즐거워하는 모습을 더 보고 싶어져서 같은 특징을 갖는 게임판을 몇 번 더 만들어 주기로 했다. 석환이는 멋진 해커이기 때문에 게임판의 규칙을 따르지 않고 타일을 하나씩만 뒤집을 수도 있다. 하지만 아무 조건 없이 타일을 뒤집는 것은 별로 재미가 없었기 때문에, 석환이는 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 ., -만으로 이루어져 있으며, #는 검은색 면이 위를 향하도록 고정된 타일, .는 흰색 면이 위를 향하도록 고정된 타일, -는 석환이가 마음대로 뒤집을 수 있는 타일을 의미한다.

Output

N 개의 줄에 걸쳐 석환이의 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 아무 것이나 출력하고, 답이 존재하지 않을 경우 `thinking_face`를 출력한다.

Sample input and output

standard input	standard output
<pre> 7 ---.--- -----.- --#----- .------ -----# -.------ ----#-- </pre>	<pre> #.#.####.# #.###### ..#.#.. #####.# #.#.... ###.#.# </pre>
<pre> 5 ----# #--.- -#--- ----. --.- </pre>	<pre> thinking_face </pre>

Problem D. 전생했더니 슬라임 연구자였던 건에 대하여 (Hard)

안녕? 내 이름은 ntopia!

나는 원래 지구에 살고있던 평범한 20대 청년이었어. 어느날 길을 걷다가 괴한의 칼에 찔려 죽어버렸어. 그런데 이게 무슨 일이란! 정신을 차려보니 이세계에 떨어져버렸지 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 될거야. 이 연구를 도와주지 않겠나?

이 곳의 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 갖고 있는 슬라임과 B 만큼의 슬라임 에너지를 갖고 있는 슬라임이 있었다고 해보자. 이 슬라임 2마리를 합성하면 슬라임 에너지가 $A * B$ 인 슬라임을 만들 수 있어.

그리고 슬라임 합성 기술이 아직 완벽하지 않아서 슬라임을 합성할 때마다 크나큰 전기 에너지를 필요로 해. 구체적으로, A 만큼의 슬라임 에너지를 갖고 있는 슬라임과 B 만큼의 슬라임 에너지를 갖고 있는 슬라임을 합성하려면 $A * B$ 만큼의 전기 에너지가 필요해.

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

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

Output

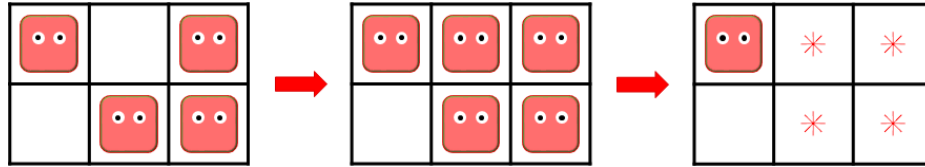
각 테스트 케이스마다 슬라임을 끝까지 합성했을 때 청구될 비용의 최소값을 1,000,000,007로 나눈 나머지를 출력한다.

Sample input and output

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

Problem E. 넴모넴모 2

넴모는 $N \times M$ 게임에 깊은 감명을 받아 “넴모넴모” 게임을 만들었다. 이 게임은 직사각형 모양의 격자판과 “넴모”라는 수수께끼의 생물을 이용해 하는 아주 간단한 게임이다. 게임의 규칙은 격자판의 비어 있는 칸을 임의로 골라 “넴모”를 하나 올려놓고, “넴모”가 올라간 칸 네 개가 2×2 사각형을 이룰 경우 그 위에 있는 “넴모”들을 모두 없애는 것을 질릴 때까지 반복하는 것이다.



하지만 안타깝게도 게임은 정말 재미가 없었고, 넴모는 아주 빨리 질려 버리고 말았다. 실망한 넴모는 격자판 위에 더 이상 없앨 “넴모”가 없는 상태에서 게임을 그만두기로 했다. 게임의 결과로 나올 수 있는 상태의 가짓수를 구하여라.

Input

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

Output

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

5×7 격자판에 “넴모”들을 안정되게 배치하는 방법은 총 11,185,495,872가지가 있다.

Problem F. 구간 합 최대

지학이는 음이 아닌 정수 N 개로 구성된 배열을 가지고 놀고 있다. 이 배열에는 M 개의 특별한 조건이 있는데, 그것은 길이가 L_i 인 연속한 구간을 잡았을 때 합이 S_i 를 넘지 않는다는 것이다. 1 이상 N 이하의 모든 정수 K 에 대해서, 가능한 모든 배열의 길이가 K 인 연속한 구간 합의 최대값을 구하여라.

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 가능한 모든 배열의 길이가 K 인 연속한 구간 합의 최대값을 출력한다.

Sample input and output

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

Notes

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

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

Problem G. 그림 그리기

지학이는 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. 각 색으로 칠해진 넓이가 얼마나 되는지 구하여라.

Input

첫 번째 줄에 N, M, K 가 주어진다. ($1 \leq N, M, K \leq 500,000$)

두 번째 줄에 각 행의 높이를 나타내는 N 개의 정수 H_i 가 공백을 사이에 두고 주어진다. ($1 \leq i \leq N, 1 \leq H_i \leq 5,000$)

세 번째 줄에 각 열의 너비를 나타내는 M 개의 정수 W_j 가 공백을 사이에 두고 주어진다. ($1 \leq j \leq M, 1 \leq W_j \leq 5,000$)

Output

L 줄에 걸쳐 k 번째 줄에 k 번 색으로 칠해진 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

지학이가 그린 그림은 다음처럼 생겼다.



Problem H. 데굴데굴

종훈이는 질 좋은 문제와 푸집한 상품이 주어지는 서울대학교 프로그래밍 경시대회에 출전했다. 하지만 참가 신청을 할 때 실수를 하는 바람에 숨겨진 난이도인 Div. 0 난이도에 신청을 하고 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 볼록다각형이고 옆면이 직사각형으로 이루어진 각기둥 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 거라고 생각했지만, 이미 너무 지쳐 있었기 때문에 대신 Div. 1 대회장에서 열심히 문제를 풀고 있는 당신에게 도움을 요청했다.

편의를 위해 물병의 높이는 1cm 이며, 물병의 물은 물병을 기울이는 즉시 평형을 되찾는다고 가정한다.

Input

첫 번째 줄에 물병의 밑면을 이루는 꼭짓점의 수 N ($1 \leq N \leq 100,000$), 물병에 들어 있는 물의 양을 의미하는 정수 W ($1 \leq W \leq$ 물병의 부피, 단위는 cm^3)가 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 반시계 방향으로 돌면서 만나는 꼭짓점들의 좌표를 순서대로 기록한 값이다. 각 좌표값은 모두 정수이며 $-100,000,000$ 이상 $100,000,000$ 이하이다.

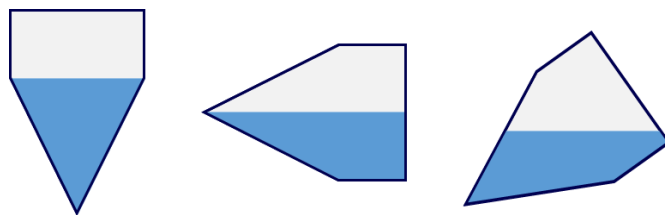
Output

첫 번째 줄에 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 18 -5 2 1 -1 4 -1 4 5 1 5	3 5

Notes



위의 그림은 물이 차지하는 영역이 삼각형, 사각형, 오각형을 이루는 경우이다.