

2025 IUPC

문제지

A.	인경호 울타리 공사	(2 pages)
B.	신묘마루의 요술망치	(3 pages)
C.	쿠키런	(2 pages)
D.	인덕이와 보드게임	(2 pages)
E.	나이트 오브 나이트	(2 pages)
F.	예쁘게 출력한 이진 트리	(2 pages)
G.	삼각형 수열	(2 pages)
H.	체크박스 누르기	(1 pages)
I.	인경호 확장판	(2 page)
J.	인덕이와 산책	(3 pages)
K.	징검다리의 징검다리	(2 pages)
L.	두 괄호 문자열	(2 pages)

주최 :



인하대학교 컴퓨터공학과
INHA UNIVERSITY

주관 :



후원 :



인 하 대 학 교
S W 중 심 대 학 사 업 단



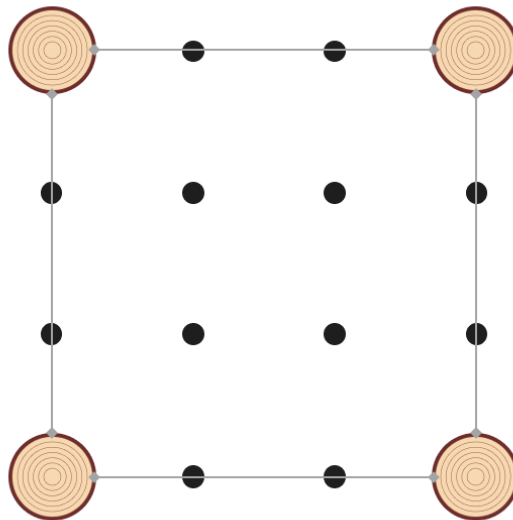
A. 인경호 울타리 공사

시간 제한 : 1초, 메모리 제한 : 1024 MB

못된 오리알 도둑이 밤마다 인경호에 몰래 침입해 인덕이들의 알을 훔쳐가기 시작했다! 못한 오리알 도둑을 막기 위해 인덕이들이 힘을 합쳐 인경호에 울타리를 설치하려 한다.

인경호는 세로 N 개, 가로 M 개의 격자점으로 이루어진 직사각형 형태의 호수이며, 상하좌우로 인접한 격자점들 간의 거리는 모두 1로 동일하다. 인덕이들은 못한 오리알 도둑을 매우 싫어하기에, 인경호의 격자점 4개를 각 꼭짓점으로 하는 **가장 큰 정사각형 형태**로 울타리를 설치하려 한다.

울타리 공사 예산이 충분하지 않기 때문에, 울타리는 인경호 바깥으로 이어질 수 없으며 인경호 전체를 감싸는 형태도 될 수 없다. 즉, 다음과 같이 울타리를 설치하는 작업은 울타리가 인경호 전체를 감싸기 때문에 불가능하다.



인덕이들이 인경호에 울타리를 설치할 때, 설치한 울타리를 이용해 만들 수 있는 가장 큰 정사각형의 넓이를 구하시오.

입력

첫 번째 줄에 정수 N , M 이 공백으로 구분되어 주어진다.

출력

인덕이들이 설치한 울타리를 이용해 만들 수 있는 가장 큰 정사각형의 넓이를 출력한다.

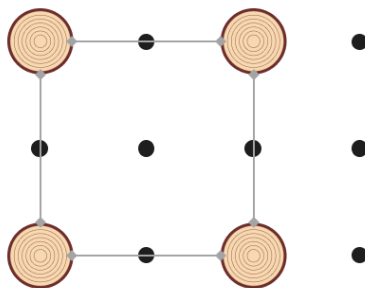
제한

• $3 \leq N \leq 1000$

• $3 \leq M \leq 1000$

예제 입력 1

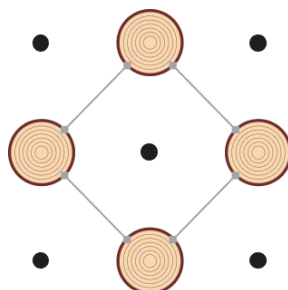
3 4	4
-----	---



예제 1에서, 가장 큰 정사각형을 만들도록 다음과 같이 울타리를 설치할 수 있다.

예제 입력 2

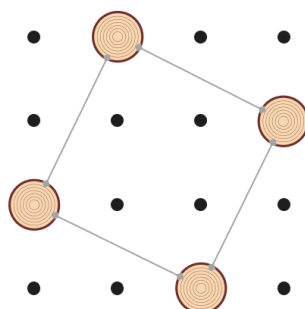
3 3	2
-----	---



예제 2에서, 가장 큰 정사각형을 만들도록 다음과 같이 울타리를 설치할 수 있다.

예제 입력 3

4 4	5
-----	---



예제 3에서, 가장 큰 정사각형을 만들도록 다음과 같이 울타리를 설치할 수 있다.

노트

인덕이는 인하대학교 캠퍼스 내부 인경호에 서식하는 귀여운 오리다.

B. 신묘마루의 요술망치

시간 제한 : 1초, 메모리 제한 : 1024 MB



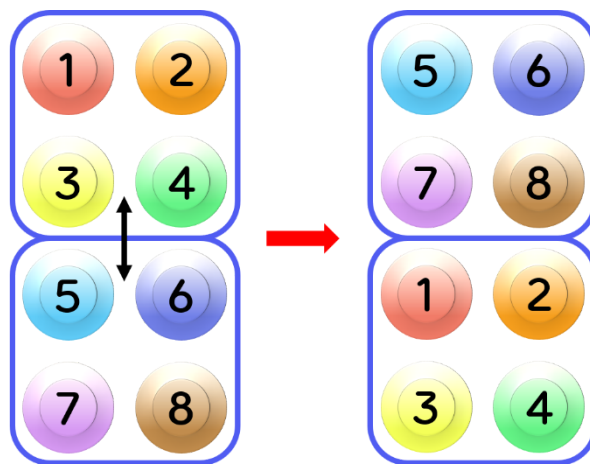
"IUPC 2025에 참가하신 여러분, 환영합니다! 대회 시작 전에 저희가 준비한 맛있는 햄버거를... 어라?"

마음씨 좋은 IUPC 운영진들은 참가자들을 위해 **맛있는 햄버거**를 잔뜩 준비했다.

하지만 신묘마루는 참가자들이 **맛있는 햄버거**를 그냥 가져가는 것이 마음에 들지 않았던 모양이다. 극악무도한 신묘마루는 **맛있는 햄버거**를 8개의 밥그릇 중 하나에 숨겨버렸다. 심지어 요술망치를 사용해서 밥그릇을 뒤죽박죽 섞어버렸다!

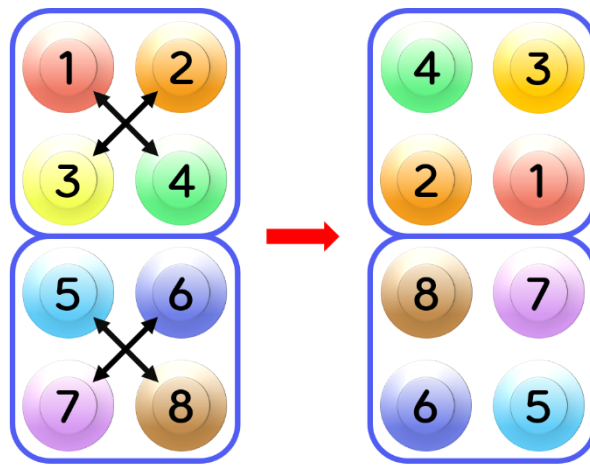
...이렇게 운영진들이 준비한 **맛있는 햄버거**는 사악한 신묘마루의 손에 들어가고 말았다. 게다가 운영진들이 밥그릇에 손도 대지 못하게 지키고 있기 때문에, 참가자들이 직접 숨겨져 있는 **맛있는 햄버거**를 찾아갈 수밖에 없게 되었다. 다행인 점은 신묘마루가 **맛있는 햄버거**를 처음에 어느 밥그릇에 넣었는지부터 시작해서 요술망치로 밥그릇을 섞는 과정까지 운영진들이 모두 찍어두었다는 것이다!

이 영상을 받아서 분석해 본 당신은 8개의 밥그릇은 처음에 4행 2열 형태로 배치되어 있으며, 신묘마루가 요술망치로 밥그릇을 섞는 기술이 아래의 4가지로 정해져 있음을 알게 되었다. 신묘마루가 사용하는 기술의 이름은 죄다 길고 복잡해서 쓰기 어려우므로, 편의상 이 기술들을 기술 A ~ 기술 D라고 부르자.



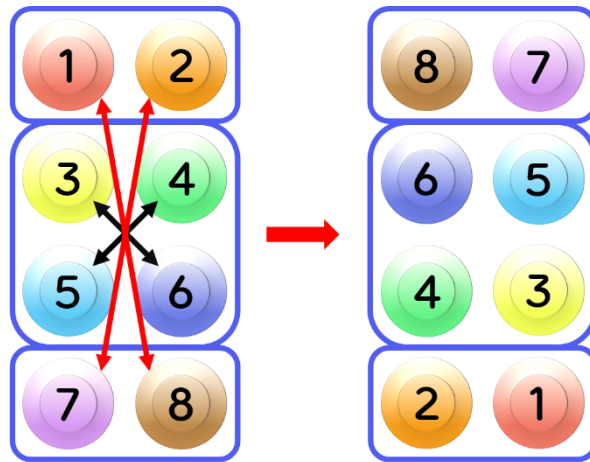
<기술 A>

기술 A: 1행과 2행에 있는 밥그릇 4개, 3행과 4행에 있는 밥그릇 4개의 위치를 그림과 같이 바꾼다.



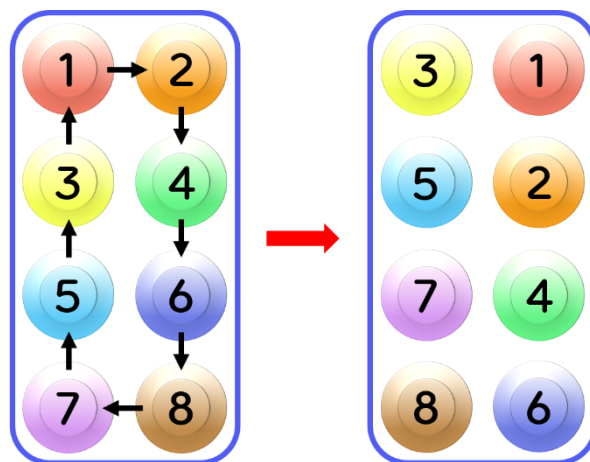
<기술 B>

기술 B: 1행과 2행에 있는 밥그릇 4개, 3행과 4행에 있는 4개의 밥그릇에서 그림과 같이 각각 X자 형태의 교환이 이루어진다.



<기술 C>

기술 C: 1행과 4행에 있는 밥그릇 4개, 2행과 3행에 있는 4개의 밥그릇에서 그림과 같이 각각 X자 형태의 교환이 이루어진다.



<기술 D>

기술 D: 8개의 밥그릇이 시계 방향으로 1칸씩 회전한다.

빈 밥그릇을 고르면 신묘마루가 요술망치(물리)로 당신을 괴롭힐 것이다. 맛있는 햄버거가 숨겨져 있는 밥그릇을 정확하게 찾을 수 있게 코드를 작성해보자!

입력

첫 번째 줄에는 맛있는 햄버거를 처음 숨긴 밥그릇의 위치 N 행 M 열의 위치 정보가 공백으로 구분되어 주어진다.

두 번째 줄에는 신묘마루가 기술을 사용한 횟수 K 가 주어진다.

세 번째 줄에는 길이 K 의 문자열 S 가 주어진다. 문자열의 i 번째 문자는 신묘마루가 i 번째로 사용한 기술을 의미한다.

출력

맛있는 햄버거가 들어있는 밥그릇의 위치의 행과 열을 공백으로 구분하여 출력한다.

제한

- $1 \leq N \leq 4$
- $1 \leq M \leq 2$
- $1 \leq K \leq 3\,000$
- 문자열 S 는 'A', 'B', 'C', 'D' 로만 이루어져 있다.

예제 입력 1

2 1 3 AAA	4 1
-----------------	-----

예제 입력 2

4 2 5 DDDDD	1 2
-------------------	-----

예제 입력 3

1 1 6 ABCDD	3 2
-------------------	-----

예제 입력 4

4 1 7 DCDBACD	4 2
---------------------	-----

C. 쿠키런

시간 제한 : 1초, 메모리 제한 : 1024 MB

과거 쿠키런 최고 권위자로 이름을 날렸던 안농이는 한때 최애 게임이었던 쿠키런이 벌써 12주년을 맞았다는 소식을 듣고 복귀 유저가 되었다!

쿠키런의 스테이지는 $3 \times N$ 크기의 2차원 배열로 구성되어 있으며, 플레이어(쿠키)는 1번 열(왼쪽 끝)에서 출발해 N 번 열(오른쪽 끝)에 도착해야 한다.



스테이지의 임의의 열에는 3×1 크기의 장애물이 있으며, 장애물은 낮은 장애물, 높은 장애물, 상단 장애물 3종류가 있다.

각 장애물은 '^', '.', 'V'의 조합으로 이루어져 있으며, 낮은 장애물은 점프를 1번 사용해, 높은 장애물은 점프를 2번 사용해, 상단 장애물은 슬라이드를 1번 사용해 각각 피할 수 있다. 점프나 슬라이드를 사용해 장애물을 회피한 후에는 직전의 점프나 슬라이드 상태가 유지되지 않는다.

출발점인 1번 열과 도착점인 N 번 열에는 장애물이 존재하지 않으며, k 번 열에 장애물이 존재한다면 바로 뒤 $k + 1$ 번 열에는 장애물이 연속하게 존재하지 않는다.

게임 시작 시점 처음 쿠키의 체력은 H 이며, 장애물의 종류와 관계없이 장애물에 1번 부딪힐 때마다 체력이 K 만큼 감소한다. 이때 체력이 0 이하가 되면 스테이지 클리어에 실패하며, 그렇지 않다면 다음 열로 이동한다. 만약 쿠키가 무사히 N 번 열에 도착한다면 스테이지 클리어에 성공한다.

쿠키런 고인물인 안농이는 발가락으로도 스테이지를 클리어할 수 있다고 자신했다. 하지만 안농이는 발가락이 짧아서 실제로 발가락으로 게임을 할 수는 없었기 때문에, 결국 고민 끝에 안농이는 점프를 J 번 이하, 슬라이드를 S 번 이하로만 사용해 스테이지를 클리어해보기로 마음먹었다.

$3 \times N$ 크기의 스테이지가 주어질 때, 안농이가 스테이지를 클리어했을 때의 최대 체력을 구해보자.

입력

첫 번째 줄에 정수 N, J, S, H, K 가 공백으로 구분되어 주어진다.

이어서 3줄에 걸쳐 스테이지의 정보가 주어진다.

출력

안농이가 점프를 J 번 이하, 슬라이드를 S 번 이하로 사용해 스테이지를 클리어했을 때의 최대 체력을 출력한다.

만약 안농이가 스테이지를 클리어할 수 없다면 대신 -1을 출력한다.

제한

$3 \leq N \leq 100$ $0 \leq J \leq 10$ $0 \leq S \leq 10$ $1 \leq H \leq 200$ $1 \leq K \leq 10$

예제 입력 1

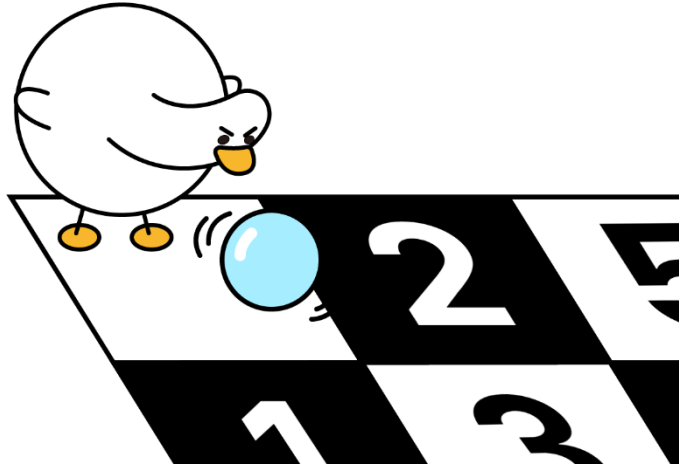
13 4 0 15 5V..... .^.^..V.....^. .^.^.....^.	5
---	---

예제 입력 2

7 0 0 1 1 ...V... ...V...	-1
--	----

D. 인덕이와 보드게임

시간 제한 : 1초, 메모리 제한 : 1024 MB



인덕이는 보드게임을 정말 좋아한다! 최근에는 안농이가 만들어준 보드게임에 푹 빠져서 헤어 나오지 못하는 중이다. 같이 이 보드게임의 설명서를 읽어보자.

<설명서>

1. 보드게임을 처음 시작하면 $N \times M$ 격자판 모양 보드의 각 칸에 임의의 정수가 하나씩 써지고, 모든 칸이 흰색 또는 검은색으로 칠해지게 됩니다. 단, 1행 1열은 항상 흰색으로 칠해집니다.
2. 이후 보드의 1행 1열에 공이 하나 소환되며 이 칸에 적혀 있는 수가 공에 기록됩니다. 이 공은 안농이의 특수한 마법이 걸려 있어 대각선 방향으로 이동할 수 없으며, 한 번에 변으로 인접한 다른 한 칸으로만 이동할 수 있습니다.
3. 공이 다른 칸으로 이동을 마치면 해당 칸에 적혀 있는 수가 공에 더해지게 됩니다. 이 때, 검은색 칸으로 이동한 경우 수가 더해진 뒤 공에 적힌 수의 부호가 반전됩니다. 흰색 칸으로 이동한 경우에는 부호 반전이 일어나지 않습니다.
4. 이제 규칙을 준수하며 공을 N 행 M 열까지 최단거리로 이동시키면 됩니다. 도착했을 때 공에 적혀 있는 수가 가능한 최댓값이라면 공이 **맛있는 너겟**으로 바뀌게 됩니다!

인덕이는 수도 없이 이 보드게임을 했지만 단 한 번도 **맛있는 너겟**을 따내지 못했다. 인덕이가 보드게임을 성공적으로 끝마치고 **맛있는 너겟**을 먹을 수 있게, 여러분들이 인덕이 대신 가능한 최댓값을 구해주자. 인덕이한테 가능한 최댓값만 알려주면, 인덕이가 공을 굴리는 방법을 찾아내는 데 큰 도움이 될 것이다.

입력

첫 번째 줄에 보드의 크기를 나타내는 두 정수 N 과 M 이 공백으로 구분되어 주어진다. N 은 보드의 행의 수, M 은 보드의 열의 수를 나타낸다.

이어서 N 개의 줄에 i 행 j 열에 적혀 있는 정수 B_{ij} 가 공백으로 구분되어 주어진다.

이어서 N 개의 줄에 보드의 i 행 j 열의 색상 정보가 공백으로 구분되어 주어진다. 0이면 흰색, 1이면 검은색으로 칠해져 있음을 의미한다. 1행 1열은 항상 흰색으로 칠해져 있음이 보장된다.

출력

공이 (N, M) 까지 최단거리로 이동하여 도착했을 때, 공에 적힐 수 있는 수의 최댓값을 출력한다.

제한

- $2 \leq N \leq 1\,000$
- $2 \leq M \leq 1\,000$
- $-10^9 \leq B_{ij} \leq 10^9$

예제 입력 1

2 2 100 20 -30 70 0 0 0 1	-140
---------------------------------------	------

예제 입력 2

2 5 2 -5 1 -3 5 3 -8 -3 5 8 0 1 0 0 0 1 1 0 1 1	7
---	---

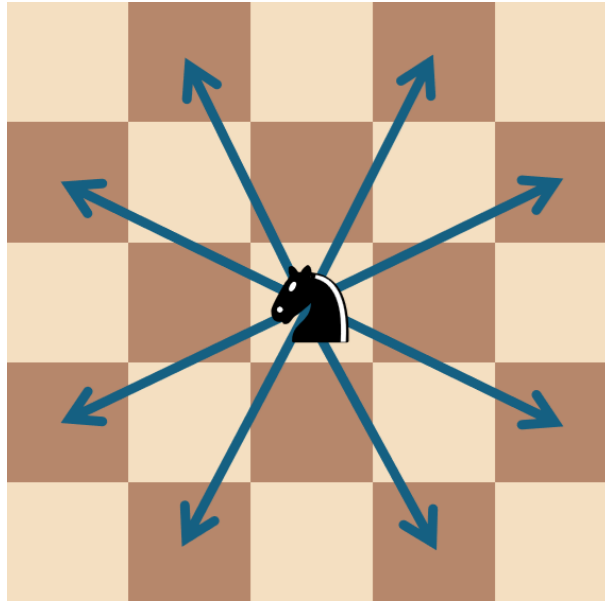
E. 나이트 오브 나이트

시간 제한 : 1초, 메모리 제한 : 1024 MB

$N \times N$ 크기의 체스판이 있다. 안즈는 이 체스판 위에 나이트를 적절하게 배치하여 최대한 많은 점수를 얻으려고 한다.

체스판의 x 행 y 열에 위치한 칸에 나이트를 놓는다면, $A_{x,y}$ 점을 얻을 수 있다. 단, 서로 공격할 수 있는 두 나이트는 동시에 배치할 수 없다.

체스에서 나이트는 'L'자 형태로 이동하며, 한 번에 가로로 두 칸 이동 후 세로로 한 칸, 또는 세로로 두 칸 이동 후 가로로 한 칸 이동할 수 있다. 아래는 나이트가 이동할 수 있는 위치를 나타낸 그림이다.



나이트는 이동할 수 있는 위치에 있는 말을 공격할 수 있으며, 두 나이트가 이러한 상대적 위치에 있을 경우 서로를 공격할 수 있다.

안즈가 얻을 수 있는 점수의 최댓값을 구하시오.

입력

첫 번째 줄에 체스판의 크기를 나타내는 N 이 주어진다.

이어서 N 개의 줄에 걸쳐, 정수 N 개가 공백으로 구분되어 주어진다. i 번째 줄의 j 번째 수는 $A_{i,j}$ 를 나타낸다.

출력

안즈가 얻을 수 있는 점수의 최댓값을 출력한다.

제한

• $1 \leq N \leq 4$

• $0 \leq A_{i,j} \leq 1000$

예제 입력 1

3 9 8 5 2 1 2 6 9 4	25
------------------------------	----

예제 입력 2

2 1000 1000 1000 1000	4000
-----------------------------	------

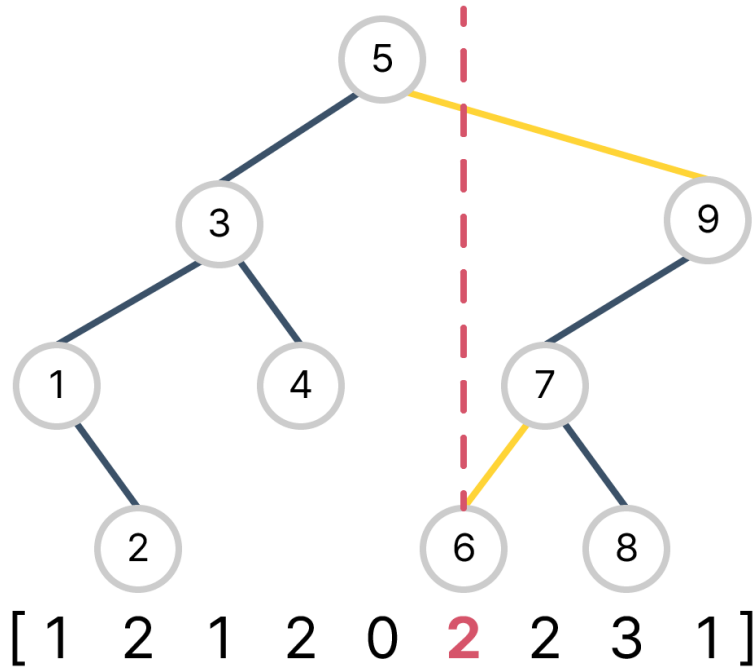
F. 예쁘게 출력한 이진 트리

시간 제한 : 1초, 메모리 제한 : 1024 MB

당신에게는 노드 개수가 N 인 이진 트리가 있다. 당신은 이 이진 트리를 중위 순회하는 순서대로 노드들의 번호를 1부터 차례대로 매겼다. 그리고 부모가 위, 자식이 아래에 위치하게끔 왼쪽에서 오른쪽으로 번호 순서대로 늘어놓았다. 아래 그림은 그렇게 이진 트리를 늘어놓은 예시이다. 이때, i 번 노드의 점수 A_i 를 다음과 같이 정의한다:

- 해당 노드의 위로 직선을 그었을 때, 해당 직선과 만나는 간선의 수 (단, 자기 자신과 부모를 잇는 간선이 존재하는 경우 그 간선도 반드시 개수에 포함한다)

노드의 개수 N 과 노드들의 점수 A_i 가 주어졌을 때, 이진 트리를 복원해 보자. 해당 점수를 갖는 이진 트리가 없는 경우는 -1을 출력한다. 답이 한 가지만 존재함을 증명할 수 있다.



입력

첫 번째 줄에 노드의 개수를 나타내는 정수 N 이 주어진다. 두 번째 줄에 각 노드의 점수를 나타내는 N 개의 정수 $A_1, A_2, \dots, A_N$ 이 공백으로 구분되어 주어진다.

출력

해당 점수를 갖는 이진 트리가 있는 경우는 첫 번째 줄에 N 개의 정수 $B_1, B_2, \dots, B_N$ 을 공백으로 구분해 출력한다.

B_i 는 i 번 노드의 부모 노드의 번호이다. i 번 노드의 부모 노드가 없는 경우 $B_i = 0$ 이다.

해당 점수를 갖는 이진 트리가 없는 경우는 첫 번째 줄에 -1을 대신 출력한다.

제한

- $1 \leq N \leq 200\,000$
- $1 \leq i \leq N$
- $0 \leq A_i \leq N - 1$

예제 입력 1

9 1 2 1 2 0 2 2 3 1	3 1 5 3 0 7 9 7 5
------------------------	-------------------

예제 입력 2

2 0 0	-1
----------	----

노트

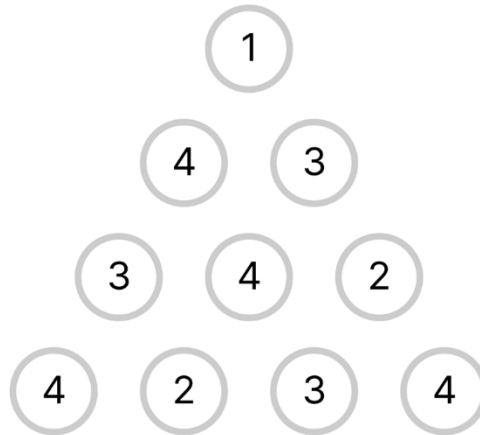
루트가 x 인 이진 트리를 중위 순회하는 순서대로 방문하는 함수 $f(x)$ 를 다음과 같이 정의할 수 있다.

1. x 의 왼쪽 자식 노드 x_l 이 존재한다면 $f(x_l)$ 을 수행한다. 존재하지 않는다면 아무것도 하지 않는다.
2. x 를 방문한다.
3. x 의 오른쪽 자식 노드 x_r 이 존재한다면 $f(x_r)$ 을 수행한다. 존재하지 않는다면 아무것도 하지 않는다.

G. 삼각형 수열

시간 제한 : 1초, 메모리 제한 : 1024 MB

당신에게는 1부터 N 까지의 정수가 있으며, 각 수 i 는 i 개씩 존재한다. 예를 들어 $N = 4$ 일 때, 1이 1개, 2가 2개, 3이 3개, 4가 4개 있다. 당신은 이 수들을 아래 그림과 같이 삼각형 모양으로 배치하려고 한다.

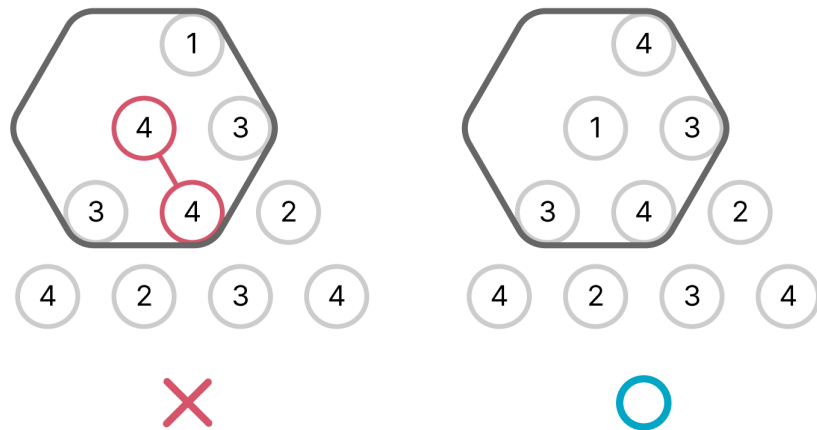


i 행의 j 번째 수를 (i, j) 로 표현했을 때, (i, j) 는 아래와 같은 조건에 의해 최대 6개의 다른 수와 인접해 있다.

- $i \geq 2$ 이고 $j \geq 2$ 라면, $(i - 1, j - 1)$ 와 인접해 있다.
- $i \geq 2$ 이고 $j \leq i - 1$ 라면, $(i - 1, j)$ 와 인접해 있다.
- $j \geq 2$ 라면, $(i, j - 1)$ 와 인접해 있다.
- $j \leq i - 1$ 라면, $(i, j + 1)$ 와 인접해 있다.
- $i \leq N - 1$ 라면, $(i + 1, j)$, $(i + 1, j + 1)$ 와 인접해 있다.

이때 모든 수들에 대해서, 수에 인접해 있는 최대 6개의 수가 해당 수와 같지 않다면 이를 올바른 삼각형 수열이라고 한다.

아래 그림에서, 왼쪽의 배치는 이를 만족하지 못하므로 올바른 삼각형 수열이 아니다. 반면, 오른쪽의 배치는 이를 만족하므로 올바른 삼각형 수열이 된다.



N 이 주어졌을 때, 올바른 삼각형 수열을 출력하라. 답이 여러 개 존재한다면 그중 아무거나 하나를 출력하라. 만약 그런 배치가 존재하지 않는다면 -1을 출력하라.

입력

첫째 줄에 정수 N 이 주어진다.

출력

N 개의 줄에 삼각형 수열을 이루는 수들을 공백으로 구분해 출력한다. i 번째 줄에는 그 줄에 속하는 i 개의 수들을 왼쪽부터 순서대로 출력한다.

만약 그런 배치가 존재하지 않는다면 대신 첫 번째 줄에 -1을 출력한다.

제한

- $1 \leq N \leq 1\,000$
- $1 \leq i \leq N$

예제 입력 1

4	4 1 3 3 4 2 4 2 3 4
---	------------------------------

예제 입력 2

3	-1
---	----

H. 체크박스 누르기

시간 제한 : 1초, 메모리 제한 : 1024 MB

길이가 1인 체크박스 N 개가 위치 0부터 중간에 빈 공간 없이 일렬로 놓여 있다. 모든 체크박스는 처음에 체크되어 있지 않은 상태이다.

체크박스가 위치 i 에 놓여 있다면, 반열린구간 $[i, i + 1)$ 을 클릭했을 때 체크박스의 상태가 바뀐다. 즉, 체크되어 있지 않았다면 체크되고, 체크되어 있었다면 체크가 해제된다.

당신은 클릭을 총 M 번 한다. 각 클릭은 위치 0부터 시작하여 $\frac{N}{M}$ 씩 떨어져 위치한다. 즉, 위치 0, 위치 $\frac{N}{M}$, 위치 $\frac{2N}{M}$, ..., 위치 $\frac{(M-1)N}{M}$ 를 각각 클릭한다. 이때, 체크된 체크박스의 수를 출력하라.

입력

첫 번째 줄에 정수 N, M 이 공백으로 구분되어 주어진다.

출력

체크된 체크박스의 수를 출력한다.

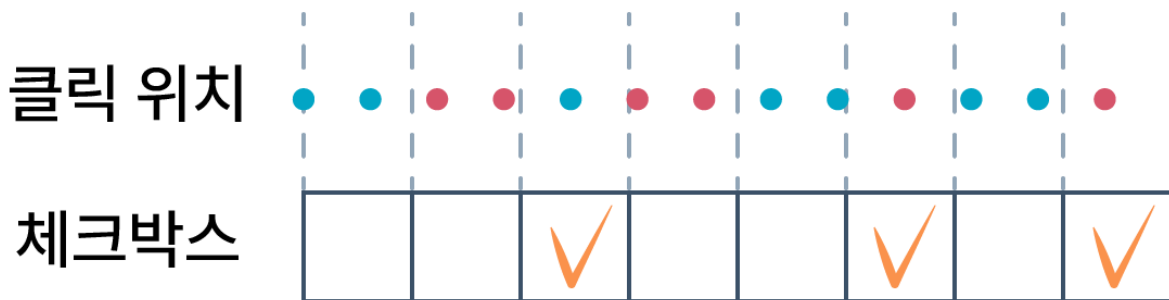
제한

$$\bullet 1 \leq N \leq 10^9$$

$$\bullet 1 \leq M \leq 10^9$$

예제 입력 1

8 13	3
------	---



노트

반열린구간 $[a, b)$ 는 a 이상 b 미만의 모든 수를 포함하는 구간이다.

I. 인경호 확장판

시간 제한 : 1초, 메모리 제한 : 1024 MB

인경호에는 인덕이와 안농이가 살고 있다. 그러나 인경호는 형태가 복잡해서 정확한 영역을 측정하기 어려웠다. 그래서 두 친구는 N 개의 꼭짓점을 가지는 단순 볼록 다각형을 인경호 둘레에 그려, 그 안쪽을 인경호의 영역으로 정했다. 그리고 N 개의 꼭짓점마다 팻말을 세워 영역을 표시했다.

그러던 어느 날, 영역이 좁다고 불평하는 인덕이를 위해 안농이는 초능력을 사용해 팻말 중 하나를 옮겨 인경호를 확장해 주기로 했다. 하지만 팻말을 너무 멀리 옮겨버리면 수상함을 감지한 인하대 학생들이 항의할 수 있기 때문에, 원래 위치에서 최대 R 만큼 떨어진 곳까지만 옮기기로 하였다.

그러나 안농이는 인경호 지하 비밀기지를 정비하느라 너무 바쁘다. 안농이 대신 인경호를 최대 넓이로 만들기 위해 옮겨야 하는 팻말의 번호와 옮겨야 하는 위치를 구해주자!

만약 최대 넓이를 만드는 것이 가능한 팻말이 여러 개라면, 안농이는 **번호가 그 중 가장 작은 팻말을 옮긴다**.

입력

첫 번째 줄에 팻말의 수 N 과 팻말을 옮길 수 있는 최대 거리인 정수 R 이 공백으로 구분되어 주어진다.

이어서 N 개의 줄에 걸쳐 두 정수 x_i, y_i 가 공백으로 구분되어 주어진다. 이는 i 번 팻말의 x 좌표와 y 좌표를 의미한다.

모든 팻말의 위치는 서로 다르며, **시계 방향 순서로 주어진다**.

출력

첫 번째 줄에 최대 넓이를 만들기 위해 옮겨야 하는 팻말의 번호를 출력한다.

두 번째 줄에 팻말을 옮겨야 하는 위치의 x 좌표와 y 좌표를 공백으로 구분하여 출력한다.

상대 오차 또는 절대 오차가 10^{-9} 이내이면 정답으로 인정된다.

제한

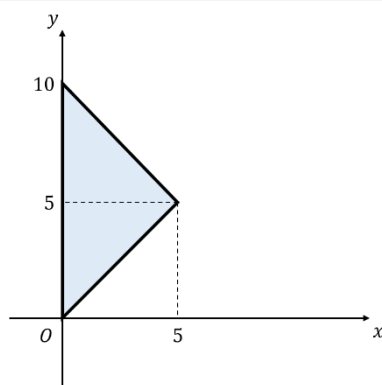
$$\bullet 3 \leq N \leq 1000$$

$$\bullet 1 \leq R \leq 10^6$$

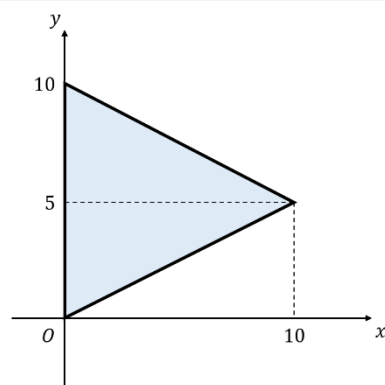
$$\bullet -10^6 \leq x_i, y_i \leq 10^6$$

예제 입력 1

3 5 0 0 0 10 5 5	3 10 5
---------------------------	-----------



[Before]

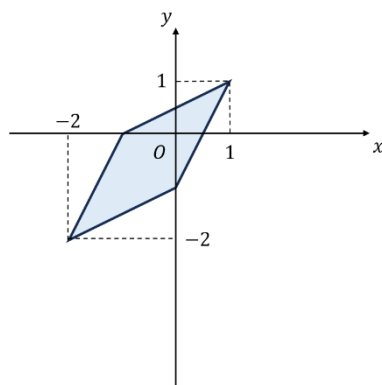


[After]

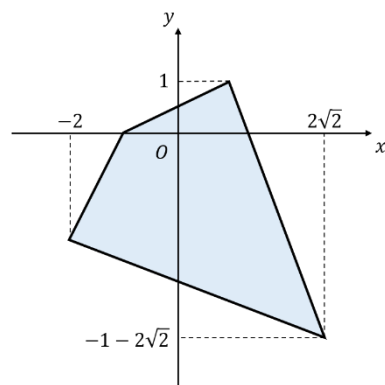
예제 1에서, 그림과 같이 3번 팻말을 $(10,5)$ 로 옮겼을 때 인경호의 넓이가 50으로 최대가 된다.

예제 입력 2

4 4 0 -1 -2 -2 -1 0 1 1	1 2.82842712474619 -3.82842712474619
-------------------------------------	---



[Before]



[After]

예제 2에서, 그림과 같이 1번 팻말을 $(2\sqrt{2}, -1 - 2\sqrt{2})$ 로 옮겼을 때 인경호의 넓이가 $3 + 6\sqrt{2}$ 로 최대가 된다.

노트

단순 볼록 다각형은 다각형을 이루는 변들이 서로 교차하지 않고 내각의 크기가 모두 180° 이하인 다각형을 말한다.

J. 인덕이와 산책

시간 제한 : 1초, 메모리 제한 : 1024 MB

인하는 N 개의 지점과 M 개의 양방향 길로 이루어져 있는 인하대의 산책로에서 산책하려고 한다. 이때, 1번 지점에서 시작해 N 번 지점에서 끝내려고 한다. 인하는 길을 따라 두 지점 사이를 이동하거나, 어떤 지점에서 이동하지 않고 1분 동안 쉴 수 있다. 두 지점 사이를 이동하는 데 걸리는 시간은 항상 1분이며, 서로 다른 두 지점 사이에는 최대 1개의 길만 존재한다. 또한, 출발지와 도착지가 같은 길은 존재하지 않는다.

한편, 인하대 내부에는 엄청 귀여운 오리인 인덕이가 살고 있다. 인덕이는 길과 무관하게 모든 임의의 두 지점 사이를 1분마다 순간 이동할 수 있다.

인덕이는 T 분 주기의 사이클을 나타내는 T 개의 번호로 이루어진 수열 c 를 따라 1분마다 산책로의 특정한 지점들 사이를 순간 이동한다. 즉, 인하가 산책을 시작한 시점으로부터 $k(k \geq 0)$ 분이 지난 시점에 인덕이는 $c_{k \bmod T}$ 번 지점에 존재한다. 이 사이클은 **항상 N 번 지점을 포함**한다. 인하가 산책을 시작하는 순간, 인덕이도 이 사이클을 따라 순간 이동을 시작한다.

만약 인하가 산책 도중에 인덕이와 마주치면 인덕이의 엄청난 귀여움에 빠져버려서 인덕이가 N 번 지점에 도착할 때까지 인덕이의 뒤에 붙어 **인덕이와 같이 순간 이동**하게 된다. 인하와 인덕이가 둘 다 1번 지점에서 시작한다면, 인하는 바로 인덕이의 뒤에 붙어버리게 된다. 인하가 N 번 정점에 도착하면 무조건 산책이 끝난다.

인하가 N 번 지점에 도착하는 데 걸리는 최소 이동 시간을 구해보자.

입력

첫 번째 줄에 지점의 수 N , 길의 수 M , 그리고 사이클의 주기 T 가 주어진다.

이어서 M 개의 줄에 걸쳐 산책로의 i 번째 길이 연결하는 두 지점의 번호 a_i, b_i 가 공백으로 구분되어 주어진다.

$M + 2$ 번째 줄에, 사이클에 포함된 지점의 번호 $c_0, c_1, \dots, c_{T-1}$ 가 공백으로 구분되어 주어진다.

출력

인하가 N 번 지점에 도착하는 데에 걸리는 최단 시간을 분 단위로 출력한다. 만약 도착할 수 없다면, -1을 대신 출력한다.

제한

- $1 \leq N \leq 1000$
- $0 \leq M \leq \min(\frac{N(N-1)}{2}, 1000)$
- $1 \leq T \leq 1000$
- $1 \leq a_i, b_i \leq N (1 \leq i \leq M)$
- $1 \leq c_j \leq N (0 \leq j < T)$

예제 입력 1

5 6 4 1 2 1 3 2 3 3 4 3 5 4 5 2 2 3 5	2
--	---

예제 입력 2

6 7 3 1 2 1 3 2 3 3 4 3 5 4 5 5 6 2 2 6	2
---	---

예제 입력 3

4 1 2 1 2 3 4	-1
---------------------	----

예제 입력 4

6 5 10 1 2 1 3 1 4 1 5 5 6 1 2 3 4 5 4 3 2 1 6	9
--	---

예제 입력 5

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

노트

정수 a 와 0이 아닌 정수 b 에 대해, $a = bq + r$ 와 $0 \leq r < |b|$ 를 만족하는 정수 q, r 이 유일하게 존재하며, 이때 r 를 a 를 b 로 나누었을 때 나머지라 한다.

$a \bmod b$ 는 a 를 b 로 나누었을 때 나머지를 의미한다.

K. 징검다리의 징검다리

시간 제한 : 1초, 메모리 제한 : 1024 MB

N 개의 호수가 일렬로 늘어서 있다. 각 호수는 왼쪽에서부터 1번에서 N 번까지 번호가 매겨져 있으며, 각 호수에는 돌들이 원형으로 배치되어 있다. i 번째 호수에는 징검다리를 이루는 돌이 A_i 개 있으며, 각 돌에는 1번부터 A_i 번까지 번호가 매겨져 있다.

용모는 S 번째 호수의 1번 돌 위에 서 있으며, 다음과 같은 규칙으로 이동한다.

- 현재 j 번 돌 위에 있다면 같은 호수의 $j + 1$ 번 돌로 이동한다. 단, 현재 A_i 번 돌 위에 있다면 1 번 돌로 이동한다.
- 현재 j 번 돌 위에 있다면 같은 호수의 $j - 1$ 번 돌로 이동한다. 단, 현재 1번 돌 위에 있다면 A_i 번 돌로 이동한다.
- 현재 i 번째 호수의 돌 위에 있다면 $i - 1$ 번 또는 $i + 1$ 번 호수의 임의의 돌로 이동한다. 단, $i - 1$ 번 호수 또는 $i + 1$ 번 호수가 존재하지 않는다면 해당 방향으로 이동할 수 없다.
- 한번 밟은 돌은 다시 밟을 수 없다.

용모가 E 번째 호수의 임의의 돌에 도착할 때까지, 시작 돌과 끝 돌을 포함하여 정확히 K 개의 돌을 밟을 수 있는지 판단하는 프로그램을 작성하시오.

입력

첫 번째 줄에 호수의 개수를 나타내는 정수 N 과 시작 호수와 끝 호수의 번호를 나타내는 정수 S, E 가 주어진다.

두 번째 줄에 각 호수의 징검다리를 이루는 돌의 개수 $A_1, A_2, \dots, A_N$ 이 공백으로 구분되어 주어진다.

세 번째 줄에 용모가 밟아야 하는 서로 다른 돌의 개수 K 가 주어진다.

출력

용모가 정확히 K 개의 돌을 밟은 후 E 번째 호수의 임의의 돌에 도착할 수 있으면 1, 그렇지 않으면 0을 출력한다.

제한

- $2 \leq N \leq 200\,000$
- $1 \leq S, E \leq N, S \neq E$
- $1 \leq A_i \leq 1\,000$
- $2 \leq K \leq 10^9$

예제 입력 1

7 6 3 4 1 7 2 2 4 9 5	1
-----------------------------	---

예제 입력 2

5 2 4 2 3 2 6 4 2	0
-------------------------	---

L. 두 괄호 문자열

시간 제한 : 1초, 메모리 제한 : 1024 MB

서기 3025년, 동현이는 3025 IUPC에 참가하였다. 마지막 문제부터 시작해서 역순으로 문제를 풀기로 결심한 동현이는 대회가 시작하자마자 가장 먼저 마지막 문제를 읽었다. 문제의 내용은 다음과 같았다.

올바른 괄호 문자열은 다음과 같이 정의된다.

1. 빈 문자열은 올바른 괄호 문자열이다.
2. A가 올바른 괄호 문자열이면, A를 괄호로 둘러싼 (A)도 올바른 괄호 문자열이다.
3. A와 B가 올바른 괄호 문자열이면, 두 문자열을 이어 붙인 AB도 올바른 괄호 문자열이다.
4. 모든 올바른 괄호 문자열은 위의 세 가지 규칙을 이용해서 만들 수 있다.

길이가 N 인 올바른 괄호 문자열 a , b 가 주어진다. 당신은 다음과 같은 시행을 원하는 만큼 할 수 있다.

- 문자열 a 의 부분문자열 $()$ 를 선택한다. 선택한 부분문자열을 $)$ (로 바꿔도 여전히 a 가 올바른 괄호 문자열이라면, $()$ (로 바꾼다.

a 를 b 와 같게 만들기 위해 필요한 최소 시행 횟수를 출력하여라. 만약 a 를 b 와 같게 만드는 것이 불가능하다면 대신 -1을 출력하여라.

문제를 읽은 동현이는 누구보다 빠르게 코드를 제출했지만 틀렸습니다 결과를 받았다. "맞는데 왜 틀려"를 반복해서 외치던 동현이는 충격적인 사실을 알아냈다. 문자열 a , b 가 입력 데이터로 주어져야 하는데 출제자의 실수로 인하여 문자열 b 만 입력 데이터로 주어진 것이다. 이래서는 문자열 a 를 모르기 때문에 문제를 풀 수가 없다.

화가 난 동현이는 길이가 N 인 가능한 모든 올바른 괄호 문자열들을 a 에 대입하여 나오는 출력값들을 모조리 더한 값을 구하는 프로그램을 제출하기로 했다. 값이 매우 커지거나 작아질 수 있으므로, 구하고자 하는 값을 $f(b)$ 라고 할 때 $f(b) \bmod 10^9 + 7$ 를 출력하기로 했다. 동현이가 제출하려는 프로그램을 여러분도 작성해 보자.

$\bmod$ 연산의 의미를 잘 모른다면 하단의 노트 문단을 참고하자.

입력

첫 번째 줄에 문자열의 길이 N 이 주어진다.

두 번째 줄에 길이가 N 인 문자열 b 가 주어진다. b 는 올바른 괄호 문자열이다.

출력

동현이가 구하고자 하는 값을 $f(b)$ 라고 할 때 $f(b) \bmod 10^9 + 7$ 를 첫 번째 줄에 출력한다.

제한

- $2 \leq N \leq 2000$

예제 입력 1

6 () ()	7
--------------	---

예제 입력 2

6 () ()	1
--------------	---

예제 입력 3

4 ()	1000000006
----------	------------

노트

정수 a 와 0이 아닌 정수 b 에 대해, $a = bq + r$ 와 $0 \leq r < |b|$ 를 만족하는 정수 q, r 이 유일하게 존재하며, 이때 r 를 a 를 b 로 나누었을 때 나머지로 한다.

$a \bmod b$ 는 a 를 b 로 나누었을 때 나머지를 의미한다.