

2022 IUPC

문제지

A.	경로당핑크 2077	(2 pages)
B.	너의 평점은	(2 pages)
C.	바벨탑의 저주	(3 pages)
D.	새벽의 탐정 게임	(2 pages)
E.	샤카샤카	(3 pages)
F.	정사각형 세기	(2 pages)
G.	조각 케이크	(1 page)
H.	크림 파스타	(1 page)
I.	타이핑	(1 page)
J.	파밍 루트	(2 pages)

주최 :  **인하대학교** INHA UNIVERSITY 주관 : 

후원 : **NAVER**  **STARTLINK** 

A. 경로당펍크 2077

시간 제한 : 1 초, 메모리 제한 : 1024 MB

시은이는 종합설계 프로젝트로 오픈월드 액션 고스톱 게임 *경로당펍크 2077* 을 개발하고 있다. 대사를 추가하던 중, 사용자 이름에 따라 '을' 또는 '를' 중 하나를 출력해야 함을 깨달았다. 예를 들어 이름이 도현이라면 도현'을' 이라고 출력해야 하고, 인하라면 인하'를' 이라고 출력해야 한다. 그 기준은 이름 마지막 글자의 받침 존재 여부이다.

초성	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
중성	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅞ	ㅟ			
종성	없음	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ

사용자는 반드시 한글 음절로만 구성된 이름을 사용한다. 이름의 모든 음절은 위 표의 초성, 중성, 종성을 순서대로 하나씩만 골라 조합하여 만들 수 있다. 예를 들어 음절 인(초성 ㅇ, 중성 ㅣ, 종성 ㄴ), 하(초성 ㅎ, 중성 ㅏ, 종성 없음)는 사용자 이름에 등장할 수 있다. 반대로 짜, 계, 와, 샴은 모두 위 표에 없는 날자가 포함되어 사용자 이름에 등장하지 않는다.

Q	W	E	R	T	Y	U	I	O	P	
ㅂ	ㅈ	ㄷ	ㄱ	ㅅ	ㅛ	ㅑ	ㅓ	ㅞ	ㅟ	ㅊ
A	S	D	F	G	H	J	K	L		
ㅁ	ㄴ	ㅇ	ㄹ	ㅎ	ㅗ	ㅕ	ㅏ	ㅑ	ㅓ	ㅕ
Z	X	C	V	B	N	M				
ㅋ	ㅌ	ㅊ	ㅍ	ㅌ	ㅍ	ㅌ				

위 그림과 같은 표준 두벌식 자판에서 사용자 이름을 영문 모드 소문자로 타이핑하여 얻는 문자열을 문자열 s 라고 하자. 문자열 s 가 주어지면 원래 사용자 이름 마지막 글자에 받침이 존재하는지 판별하는 프로그램을 작성하라.

입력

첫 번째 줄에 문자열 s 의 길이를 나타내는 정수 N 이 주어진다.

두 번째 줄에 문자열 s 가 주어진다.

출력

사용자 이름 마지막 글자에 받침이 있다면 1, 없다면 0 을 출력한다.

제한

- $2 \leq N \leq 30$

예제 입력 1

7 wnehgus	1
--------------	---

사용자 이름은 주도현이다.

예제 입력 2

8 rlarudxo	0
---------------	---

사용자 이름은 김경태다.

예제 입력 3

10 anzufutaba	1
------------------	---

사용자 이름은 무켜럿뒸이다.

노트

현대 한글의 모든 초성, 중성, 종성 목록은 다음과 같다.

- 초성 : ㄱ, ㅋ, ㄴ, ㄷ, ㄸ, ㄹ, ㄱ, ㅁ, ㅂ, ㅃ, ㅅ, ㅆ, ㅇ, ㅈ, ㅊ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ
- 중성 : ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ
- 종성 : (없음), ㄱ, ㅋ, ㄴ, ㄷ, ㄸ, ㄹ, ㄱ, ㅁ, ㅂ, ㅃ, ㅅ, ㅆ, ㅇ, ㅈ, ㅊ, ㅊ, ㅋ, ㅌ, ㅍ, ㅎ

하지만 이 문제에선 편의를 위해 입력으로 주어질 수 있는 초성, 중성, 종성의 종류를 제한하였다.

B. 너의 평점은

시간 제한 : 1 초, 메모리 제한 : 1024 MB

인하대학교 컴퓨터공학과를 졸업하기 위해서는, 전공평점이 3.3 이상이거나 졸업고사를 통과해야 한다. 그런데 아뿔싸, 치훈이는 깜빡하고 졸업고사를 응시하지 않았다는 사실을 깨달았다!

치훈이의 전공평점을 계산해주는 프로그램을 작성해보자.

전공평점은 전공과목별 (학점 × 과목평점)의 합을 학점의 총합으로 나눈 값이다.

인하대학교 컴퓨터공학과에 따른 과목평점은 다음 표와 같다.

A+	4.5
A0	4.0
B+	3.5
B0	3.0
C+	2.5
C0	2.0
D+	1.5
D0	1.0
F	0.0

P/F 과목의 경우 등급이 P 또는 F로 표시되는데, 등급이 P인 과목은 계산에서 제외해야 한다.

과연 치훈이는 무사히 졸업할 수 있을까?

입력

20 줄에 걸쳐 치훈이가 수강한 전공과목의 과목명, 학점, 등급이 공백으로 구분되어 주어진다.

출력

치훈이의 전공평점을 출력한다.

정답과의 절대오차 또는 상대오차가 10^{-4} 이하이면 정답으로 인정한다.

제한

- $1 \leq$ 과목명의 길이 ≤ 50
- 과목명은 알파벳 대소문자 또는 숫자로만 이루어져 있으며, 띄어쓰기 없이 주어진다. 입력으로 주어지는 모든 과목명은 서로 다르다.
- 학점은 1.0, 2.0, 3.0, 4.0 중 하나이다.
- 등급은 A+, A0, B+, B0, C+, C0, D+, D0, F, P 중 하나이다.
- 적어도 한 과목은 등급이 P가 아님이 보장된다.

예제 입력 1

ObjectOrientedProgramming1 3.0 A+ IntroductiontoComputerEngineering 3.0 A+ ObjectOrientedProgramming2 3.0 A0 CreativeComputerEngineeringDesign 3.0 A+ AssemblyLanguage 3.0 A+ InternetProgramming 3.0 B0 ApplicationProgramminginJava 3.0 A0 SystemProgramming 3.0 B0 OperatingSystem 3.0 B0 WirelessCommunicationsandNetworking 3.0 C+ LogicCircuits 3.0 B0 DataStructure 4.0 A+ MicroprocessorApplication 3.0 B+ EmbeddedSoftware 3.0 C0 ComputerSecurity 3.0 D+ Database 3.0 C+ Algorithm 3.0 B0 CapstoneDesigninCSE 3.0 B+ CompilerDesign 3.0 D0 ProblemSolving 4.0 P	3.284483
--	----------

예제 입력 2

BruteForce 3.0 F Greedy 1.0 F DivideandConquer 2.0 F DynamicProgramming 3.0 F DepthFirstSearch 4.0 F BreadthFirstSearch 3.0 F ShortestPath 4.0 F DisjointSet 2.0 F MinimumSpanningTree 2.0 F TopologicalSorting 1.0 F LeastCommonAncestor 2.0 F SegmentTree 4.0 F EulerTourTechnique 3.0 F StronglyConnectedComponent 2.0 F BipartiteMatching 2.0 F MaximumFlowProblem 3.0 F SuffixArray 1.0 F HeavyLightDecomposition 4.0 F CentroidDecomposition 3.0 F SplayTree 1.0 F	0.000000
---	----------

노트

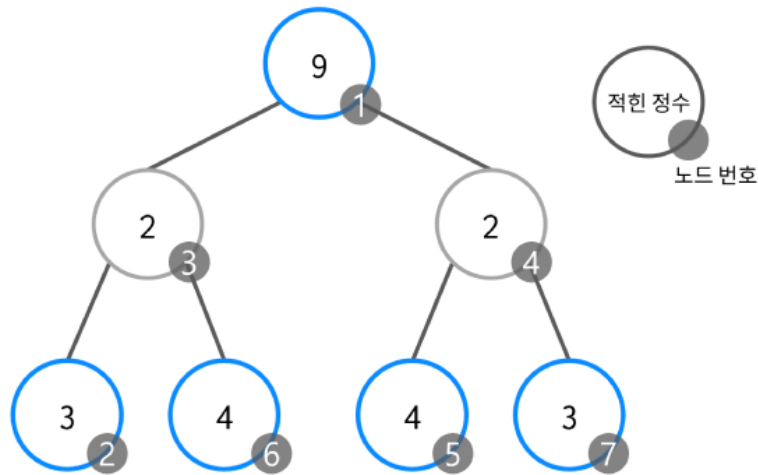
예제 1 은 치훈이의 실제 전공과목 성적이다.

C. 바벨탑의 저주

시간 제한 : 2 초, 메모리 제한 : 1024 MB

바벨탑은 N 개의 방이 $N-1$ 개의 통로로 연결된 이차원 건축물이다. 어떤 방들은 저주의 방으로, 접근하면 화를 입는다고 한다.

탑은 아래 그림과 같이 방과 통로가 각각 노드와 간선인 트리 그래프로 도식화할 수 있다. 각 노드에는 1 이상 N 이하의 고유한 노드 번호가 있고, 번호가 i 인 노드에는 중복될 수 있는 정수 a_i 가 적혀있다. 루트 노드의 번호는 항상 1 번이다. 한 부모 노드에게 자식 노드가 여럿이라면 번호가 더 작은 자식이 왼쪽에 있다.



번호가 i 인 노드를 루트 노드로 하며 그 노드의 부모 노드를 포함하지 않는 가장 큰 서브트리를 T_i 라고 하자. i 번 노드를 좌우로 뒤집으면 T_i 에 속하는 모든 노드에 대해, 반대로 노드 번호가 더 큰 자식이 왼쪽에 있게 된다.

저주 노드란 좌우로 뒤집어도 전체 트리의 구조와 노드에 적힌 정수가 원래와 일치하는 노드이다. 위 그림에서 저주 노드들은 파란 테두리로 표시되어 있다. 당신의 임무는 모든 **저주 노드**의 번호를 알아내는 것이다.

입력

첫째 줄에 노드의 수 N 이 주어진다.

둘째 줄에 a_1 부터 a_N 까지의 N 개의 정수가 공백으로 구분되어 주어진다.

다음 $N - 1$ 개의 줄에 걸쳐 두 정수 i, j 가 공백으로 구분되어 주어진다. i 번 노드가 j 번 노드의 부모라는 뜻이다.

출력

첫째 줄에 저주 노드의 수를 출력한다.

둘째 줄에 저주 노드들의 노드 번호를 작은 수부터 공백으로 구분하여 출력한다.

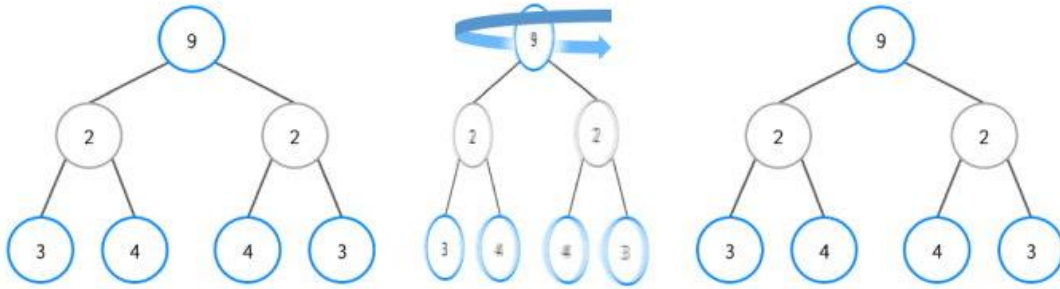
제한

- $1 \leq N \leq 500,000$
- $-10^9 \leq a_i \leq 10^9$

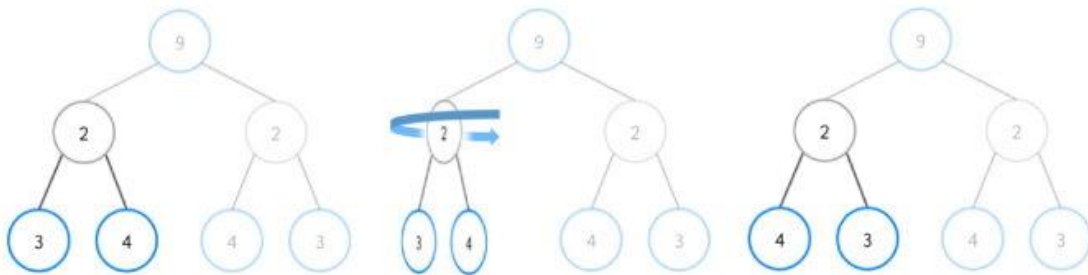
예제 입력 1

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

1 번 노드는 좌우로 뒤집었을 때 처음과 같은 모습이므로 저주 노드이다.



3 번 노드는 좌우로 뒤집기 전과 후의 트리의 구조는 같지만 노드에 적힌 정수가 달라지므로 저주 노드가 아니다.

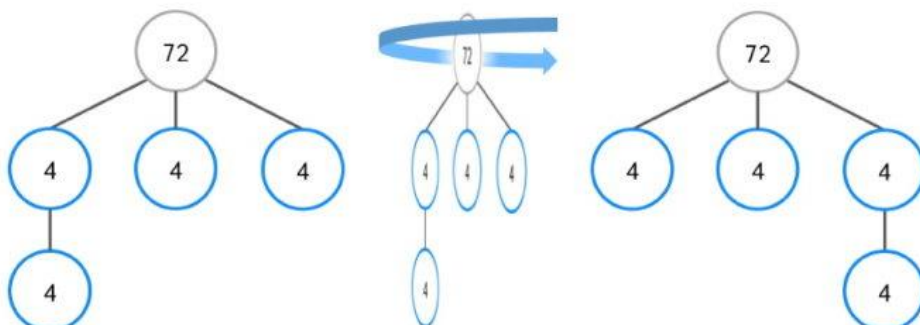


특히, 정의에 의해 리프 노드는 반드시 저주 노드이다.

예제 입력 2

5 72 4 4 4 4 1 2 1 3 1 4 2 5	4 2 3 4 5
---	--------------

1 번 노드는 좌우로 뒤집기 전과 후의 트리의 구조가 달라 저주 노드가 될 수 없다.



예제 입력 3

13 3 5 2 9 6 8 4 4 1 6 5 8 8 1 2 1 3 1 11 11 10 11 8 11 7 11 5 7 4 7 9 2 6 6 12 2 13	9 3 4 5 6 8 9 10 12 13
---	---------------------------

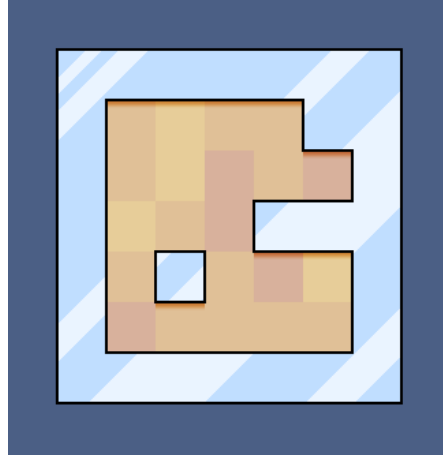
예제 입력 4

1 -3	1 1
---------	--------

D. 새벽의 탐정 게임

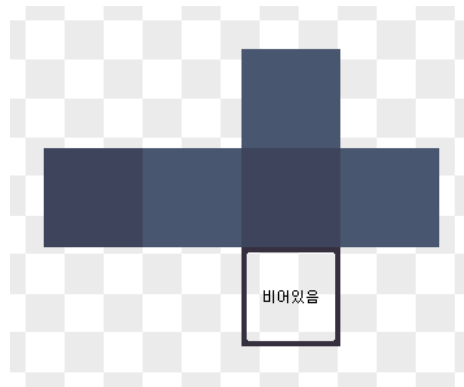
시간 제한 : 1 초, 메모리 제한 : 1024 MB

새벽의 탐정 게임은 재미있는 2 인용 보드게임이다. 한 명은 탐정, 다른 한 명은 도둑 역할을 맡는다.

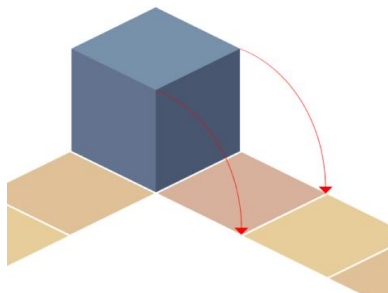


게임은 N 행 M 열 격자 위에서 진행된다. 격자의 각 칸은 벽이 설치되어 있거나 비어있고, 역할에 관계없이 벽 위에 서거나 벽을 넘어갈 수 없다. 임의의 두 빈칸은 상하좌우로 인접한 빈칸들을 거쳐 이동할 수 있다. 다시 말해 어느 빈칸에서 출발해도 다른 모든 빈칸으로 이동할 수 있다.

도둑은 밤 동안 은밀하게 어떤 칸으로 이동하고, 다시 밤이 찾아오기 전엔 그 칸에서 움직일 수 없다. 탐정은 새벽이 밝아오면 단서를 모아 도둑의 위치를 밝혀내고 감옥에 가두어야 한다.



감옥은 위 전개도를 접은 모습이다. 다시 말해, 다섯 면이 막혀있고 나머지 한 면이 뚫려있는 정육면체이다.



탐정은 자신이 있는 칸에 감옥의 뚫린 면이 바닥을 향하게 하여 놓는다. 감옥은 상하좌우 중 벽이 없는 한 방향으로 굴릴 수 있고, 그에 따라 바닥을 향하는 면이 바뀌게 된다. 즉, 이동하려는 칸의 바닥과 수직으로 접하고 있는 면이 이동 후 바닥을 향하게 된다.

도둑을 감옥에 가두려면, 감옥의 뚫린 면이 바닥을 향할 때 도둑이 같은 칸에 있어야 한다. 만약 막힌 면이 바닥을 향할 때 도둑이 같은 칸에 있다면 도둑은 바로 승리를 선언하고 탐정은 패배하게 된다.

당신은 탐정 역할이고, 도둑의 위치를 특정했다. 이제 도둑을 감옥으로 가두기만 하면 승리할 수 있다. 당신의 위치와 도둑의 위치가 주어질 때, 감옥을 최소 몇 번 굴려야 당신이 게임에서 승리할 수 있을지 알아보자.

입력

첫 번째 줄에 보드의 크기 N, M 이 주어진다.

두 번째 줄부터 N 개의 줄에 각각 M 개의 문자로 보드의 상태가 주어진다. 벽이 있는 칸은 #, 빈칸은 ., 탐정의 위치는 D, 도둑의 위치는 R로 주어진다.

출력

탐정이 승리할 수 있다면 도둑을 가두는데 필요한 최소 이동 횟수를 출력한다.

그렇지 않다면 -1을 출력한다.

제한

- $4 \leq N, M \leq 500$
- 입력에 D와 R은 한 개씩만 주어진다.
- 탐정이 있는 칸과 도둑이 있는 칸은 모두 벽이 아니다.
- 보드의 테두리, 즉 1행, 1열, N 행, M 열에 해당하는 칸은 벽이다.

예제 입력 1

<pre> 5 5 ##### #D..# #.#.# #..R# ##### </pre>	<pre> 4 </pre>
--	----------------

예제 입력 2

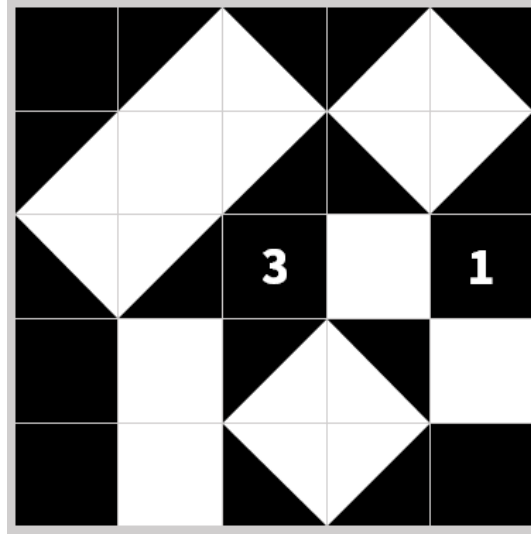
<pre> 6 5 ##### #...# #..R# ##.## ##D## ##### </pre>	<pre> 7 </pre>
--	----------------

예제 입력 3

<pre> 4 6 ##### #R..D# #...# ##### </pre>	<pre> -1 </pre>
---	-----------------

E. 샤카샤카

시간 제한 : 1 초, 메모리 제한 : 1024 MB



샤카샤카는 직사각형 보드 위에서 즐기는 퍼즐 게임이다. 보드는 N 행 M 열의 격자로 구분되어 있으며, 각 칸에 블록을 하나 놓을 수 있다. 퍼즐을 해결하려면 게임을 시작하기 전 주어지는 보드의 블록들을 적절히 교체하여 정답 조건을 만족시켜야 한다.

샤카샤카의 블록은 다음과 같은 여섯 종류이다. 앞의 2개는 정사각형 블록, 뒤의 4개는 삼각형 블록이라고 부르자.



또한, 검은색 정사각형 블록 위에는 다음과 같이 0 이상 4 이하의 숫자가 적혀있을 수 있다.



초기 보드의 각 칸에는 정사각형 블록이 놓여있다. 플레이어는 흰색 정사각형 블록을 원하는 삼각형 블록으로 바꿀 수 있으며, 정답 조건을 만족시키려면 다음 내용을 모두 만족해야 한다.

- 연속한 모든 흰색 영역은 직사각형이다. 단, 그 변이 가로축 또는 세로축과 평행할 필요는 없다.
- 숫자 k 가 적힌 정사각형 블록의 상하좌우 4칸에는 삼각형 블록이 정확히 k 개 있어야 한다.

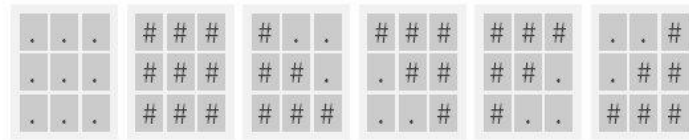
샤카샤카의 일반적인 해법을 구하는 것은 NP-Complete 문제로 아주 어렵다는 것이 잘 알려져 있다. 퍼즐을 푸는 대신 주어진 샤카샤카 해답이 정답 조건을 만족하는지 판별하는 프로그램을 작성하여라.

입력

입력의 첫 번째 줄에는 보드의 크기 N , M 이 주어진다.

다음 $3N$ 개의 줄에는 샤카샤카의 해답이 다음과 같은 조건으로 주어진다.

본문의 6가지 블록은 다음과 같은 3×3 아스키 아트로 주어진다.



또한 검은색 정사각형 블록 안에 숫자가 있는 경우, 다음과 같이 해당 블록의 가운데에 숫자가 위치한다.



샤카샤카의 해답은 $3N \times 3M$ 개의 문자로 주어지며, 위와 같이 3×3 단위로 끊어서 어떤 블록이 어느 위치에 있는지 해석할 수 있다.

출력

주어진 해답이 정답 조건을 만족하면 YES, 아니면 NO 를 출력한다.

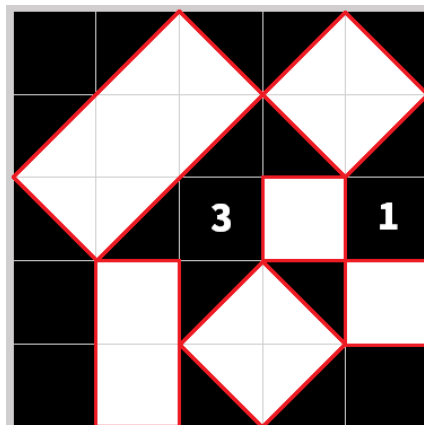
제한

- $1 \leq N, M \leq 10$

예제 입력 1

<pre> 5 5 ##### #####.###.## #####.###.### ###.....### ###.....###.### ##.....###.### #.....##### #.....##### ##.###3#...#1# #####.### ###...##### ###...###.### ###...#...### ###...#...##### ###...###.##### ###...##### </pre>	YES
---	-----

해당 예제는 아래 그림과 같이 모든 흰색 영역이 직사각형이고, 숫자가 적힌 블록 상하좌우로 알맞은 개수의 삼각형 블록이 있으므로 정답 조건을 만족한다.



예제 입력 2

5 5	NO
##### ##.##.##.## #...#...#...# #...##### ##.##### ##### ##### #####.## #####.##.##3# #####.##.## ##### #####.## ##### #...##### ##.##3### #####	

예제 입력 3

5 5#####...##.##...#. ...#... #####.....### #2###.....## ####.....# ###.....# ##.....## #.....### #.....#... ##.....##... ###.....###... ...#.....####... ...##.#####... ...#####...	YES
---	-----

예제 입력 4

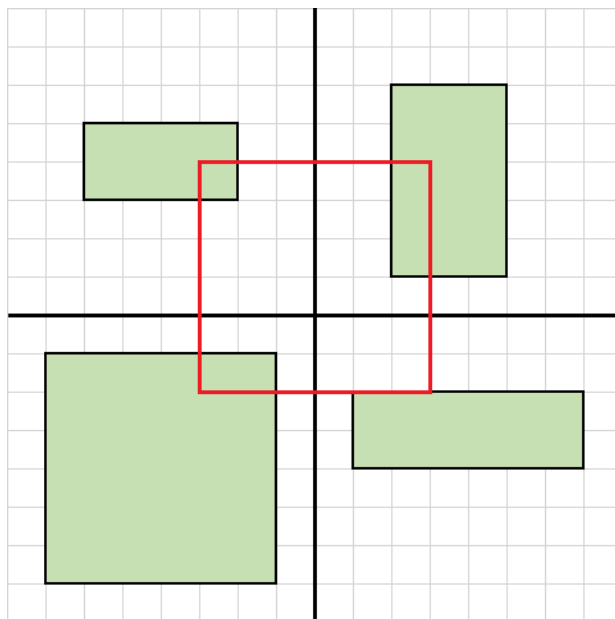
4 3	NO
.....	
.....	
.....	
...###...	
...###...	
...###...	
...###...	
...###...	
...###...	
.....	
.....	
.....	

예제 입력 5

2 3####0####	NO
--	----

F. 정사각형 세기

시간 제한 : 1 초, 메모리 제한 : 1024 MB



민겸이는 4 개의 서로 다른 사분면 위에 각 변이 x 축 또는 y 축에 평행하면서 네 꼭짓점이 모두 격자점 위에 있는 직사각형을 하나씩 그렸다. 각 직사각형에서 x, y 좌표가 모두 정수인 점들을 하나씩 선택해 만들 수 있는 정사각형의 개수를 구하라. 이때, 정사각형의 각 변은 x 축 또는 y 축에 평행해야 한다.

입력

입력은 4 줄로 주어진다. 각 줄에는 4 개의 정수 x_1, y_1, x_2, y_2 가 공백으로 구분되어 주어진다. i 번째 줄에 주어진 (x_1, y_1) 은 제 i 사분면에 그린 직사각형의 꼭짓점 중 원점에 가장 가까운 점의 좌표이고, (x_2, y_2) 은 제 i 사분면에 그린 직사각형의 꼭짓점 중 원점에 가장 먼 점의 좌표이다.

출력

각 직사각형에서 x, y 좌표가 모두 정수인 점들을 하나씩 선택해 만들 수 있는 정사각형의 개수를 출력한다.

제한

- $1 \leq |x_1|, |y_1|, |x_2|, |y_2| \leq 10^5$
- $|x_1 - x_2| \geq 1$
- $|y_1 - y_2| \geq 1$

예제 입력 1

2 1 5 6 -2 3 -6 5 -1 -1 -7 -7 1 -2 7 -4	31
--	----

예제 입력 2

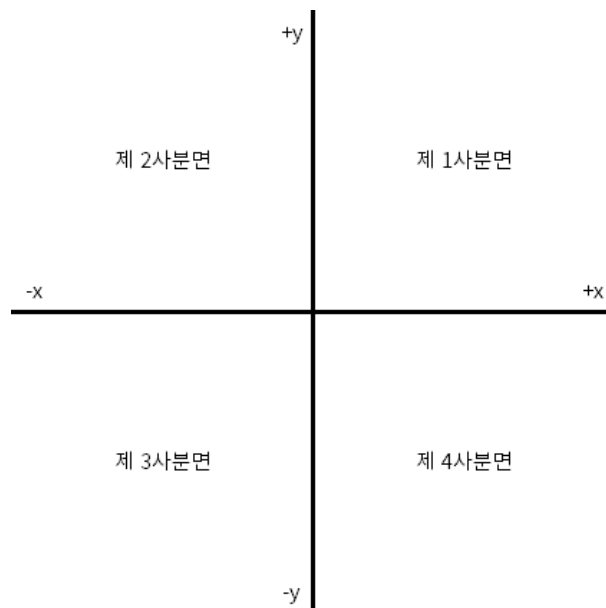
2 6 8 7 -3 2 -4 9 -1 -3 -10 -4 5 -2 6 -9	4
---	---

예제 입력 3

1 1 3 3 -2 2 -4 4 -5 -4 -7 -5 1 -1 3 -3	0
--	---

힌트

각 사분면은 아래 그림과 같이 위치한다.



G. 조각 케이크

시간 제한 : 1 초, 메모리 제한 : 1024 MB

안즈는 생일선물로 조각 케이크를 잔뜩 받았다.



케이크를 먹고 행복해하던 안즈는, 남은 케이크를 정리해 냉장고에 넣는 귀찮은 일을 해야 한다는 사실을 깨닫고 말았다.

결국 안즈는 케이크를 정리하는 좋은 방법을 알아냈다. 남은 조각 케이크들을 모아 케이크 한 판으로 만들어 버리는 것이다! 조각 케이크들의 크기의 합이 $\frac{99}{100}$ 판 이상 $\frac{101}{100}$ 판 이하라면 모아서 케이크 한 판으로 만들 수 있다.

그런데 안즈는 문득, 조각 케이크들을 골라서 한 판으로 만들 수 있는 경우의 수가 몇 가지나 되는지 궁금해졌다.

하지만 그 순간 안즈는 매우 귀찮아졌기 때문에, 여러분에게 해결을 부탁했다.

입력

첫째 줄에 남은 조각 케이크의 개수 N 이 주어진다.

둘째 줄에 각 조각 케이크의 크기를 나타내는 정수 c_i 가 공백으로 구분되어 주어진다. 이는 i 번째 조각 케이크의 크기가 $\frac{1}{c_i}$ 판임을 뜻한다.

출력

조각 케이크들을 골랐을 때, 고른 조각 케이크를 **모두** 합쳐서 케이크 한 판으로 만들 수 있는 경우의 수를 출력한다.

제한

- $1 \leq N \leq 10$
- $1 \leq c_i \leq 25$

예제 입력 1

6 3 4 6 7 9 10	2
-------------------	---

예제 입력 2

10 2 2 4 4 4 4 8 8 8 8	94
---------------------------	----

H. 크림 파스타

시간 제한 : 1 초, 메모리 제한 : 1024 MB

민겸이는 크림 파스타를 먹다가 다음과 같은 문제를 생각해냈다.

빈 배열 A 가 있다. 민겸이는 A 의 맨 뒤에 정수를 N 번 추가하려고 한다. 수를 그냥 추가하기만 하면 재미없으니, 수를 추가할 때마다 $1 \leq i \leq j \leq |A|$ 를 만족하는 정수 i, j 에 대하여 $A_j - A_i$ 의 최댓값을 구하려고 한다. $|A|$ 는 배열 A 의 현재 길이를 뜻하고, A_i 는 민겸이가 i 번째로 추가한 정수를 뜻한다.

민겸이가 식사를 마치기 전에 이 문제를 대신 풀어보자.

입력

입력은 두 줄로 주어진다.

첫 번째 줄에는 민겸이가 배열에 추가하려는 정수의 개수 N 이 주어진다.

두 번째 줄에는 A_1 부터 A_N 까지 N 개의 정수가 공백으로 구분되어 주어진다.

출력

각 A_i 가 추가된 직후의 문제의 답 N 개를 공백으로 구분하여 출력한다.

제한

- $1 \leq N \leq 200,000$
- $1 \leq A_i \leq 10^9$

예제 입력 1

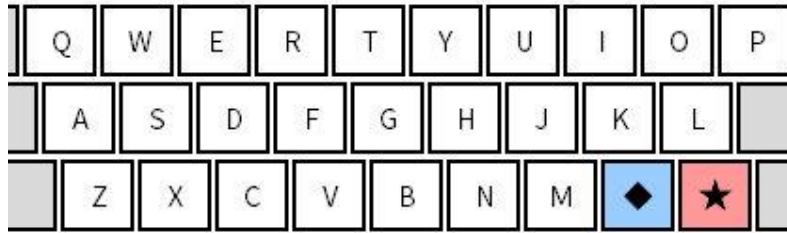
6 50 100 70 110 10 100	0 50 50 60 60 90
---------------------------	------------------

예제 입력 2

6 3 3 2 8 3 1000000000	0 0 0 6 6 999999998
---------------------------	---------------------

I. 타이핑

시간 제한 : 1 초, 메모리 제한 : 1024 MB



민겸이는 영어 소문자와 대문자로 이루어진 문자열을 타이핑하기로 했다. 민겸이가 사용할 수 있는 버튼은 26 개의 영어 알파벳 버튼과 마름모(◆) 버튼, 별(★) 버튼이다. 각 버튼은 아래와 같이 작동한다.

- 알파벳 버튼을 누르면 소문자 또는 대문자 중 하나가 입력된다. 이 때, 마름모 버튼이 활성화되어있다면 대문자, 아니라면 소문자가 입력된다. 마름모 버튼은 처음에 비활성화되어있다.
- 마름모 버튼은 한번 누를 때마다 활성화 및 비활성화 여부가 바뀐다.
- 별 버튼을 누르면 마지막으로 입력한 알파벳의 대소문자 여부가 바뀐다. 예를 들어 대문자가 마지막으로 입력되었을 경우 소문자로 바뀌고, 소문자가 마지막으로 입력되었을 경우 대문자로 바뀐다. 만약 마지막으로 입력한 알파벳이 없다면 작동하지 않는다.

민겸이는 사용할 수 있는 28 개의 버튼을 이용해 어떤 문자열을 입력하려고 한다. 이때, 가능한 한 적은 횟수만큼 버튼을 누르고 싶다. 민겸이가 해당 문자열을 입력하기 위해 버튼을 최소 몇 번 눌러야 하는지 알려주자.

입력

입력은 한 줄로 주어진다.

첫 번째 줄에 민겸이가 타이핑할 문자열이 주어진다.

출력

민겸이가 해당 문자열을 입력하기 위해 버튼을 최소 몇 번 눌러야 하는지 출력한다.

제한

- $1 \leq \text{문자열의 길이} \leq 3,000$
- 주어지는 문자열은 알파벳 대소문자로만 이루어져 있다.

예제 입력 1

iLoveINHA	11
-----------	----

I L ★ O V E ◆ I N H A 순으로 입력하면 버튼을 11 번 누르고 입력할 수 있다.

10 번 이하의 입력으로 이 문자열을 입력하는 방법은 없다.

예제 입력 2

ConquerThePlanet	19
------------------	----

J. 파밍 루트

시간 제한 : 3 초, 메모리 제한 : 1024 MB

경태는 인하대학교에서 유행하는 게임 Conquer The Planet(이하 CTP)에 빠져있다. CTP 를 열심히 플레이하던 어느 날, 경태는 게임 속에 숨겨져 있던 던전을 발견했다! 던전 입구에는 플레이어를 위한 설명서가 붙어있었다.

설명서의 내용은 다음과 같다.

- 던전은 한 번만 입장할 수 있다.
- 던전에는 N 개의 구역과 M 개의 포탈이 존재한다. 각 구역은 1 번부터 N 번까지 번호가 매겨져 있다. 던전에 입장하면 1 번 구역에서 시작한다. 포탈은 출발 구역과 도착 구역의 쌍으로 구성된다. 출발 구역에 위치한 포탈을 통해 도착 구역으로 순간이동할 수 있다. 단, 어떤 구역에도 도착 구역이 같은 서로 다른 두 포탈은 존재하지 않는다. 순간이동 시 소요되는 시간은 없다.
- 던전에 머무를 수 있는 제한 시간 T 분이 지나면 던전에서 퇴장당한다.
- 구역마다 몬스터가 한 마리씩 존재한다. 몬스터는 플레이어가 포탈을 사용하는 것을 방해한다. 플레이어가 어떤 구역에 있는 포탈을 사용하려면 먼저 전투를 통해 해당 구역의 몬스터를 쓰러트려야 한다.
- 전투는 몬스터와 플레이어가 번갈아가면서 공격하는 것으로 이뤄진다. 선공은 항상 몬스터이다. 한쪽의 체력이 0 이하가 되어 쓰러지면 전투가 종료된다. 모든 전투는 시작부터 종료까지 항상 1 분이 소요된다.
- 몬스터의 기본 공격력을 a , 시간당 공격력 증가량을 x , 플레이어 방어력당 공격력 감소량을 y 라고 하자. 던전 입장 후 t 분이 지난 시점에 시작한 전투에서, 방어력이 d 인 플레이어가 몬스터의 공격을 받으면 체력이 $\max(0, a + xt - yd)$ 만큼 감소한다.
- 전투에서 플레이어가 몬스터를 쓰러트릴 경우, 플레이어는 몬스터가 떨어트리는 금화를 획득한다. 몬스터를 쓰러트림과 동시에 던전의 제한 시간이 끝나도 금화를 획득할 수 있다. 몬스터는 플레이어가 구역을 떠난 직후 제자리에서 부활한다.
- 전투에서 플레이어가 쓰러질 경우, 플레이어는 던전에서 퇴장당한다.

경태는 강한 것이 좋기 때문에 공격력에 올인하기로 했다. 따라서 어떤 몬스터든 일격에 쓰러트릴 수 있다. 그러나 이에 대한 대가로 경태의 체력은 단 1 에 불과하다. 던전에서 최대한 많은 금화를 모으기 위해 경태에게 필요한 방어력의 최소값을 구해보자.

입력

첫째 줄에 구역의 개수 N , 포탈의 개수 M , 제한 시간 T 가 주어진다.

이어서 N 개의 줄에 걸쳐 i 번 구역에 있는 몬스터의 정보 a_i, x_i, y_i, c_i 가 주어진다. 각각 몬스터의 기본 공격력, 시간당 공격력 증가량, 플레이어 방어력당 공격력 감소량, 떨어트리는 금화의 개수를 의미한다.

이어서 M 개의 줄에 걸쳐 포탈의 출발 구역 u 와 도착 구역 v 가 주어진다. 이는 u 번 구역에 위치한 포탈을 통해 v 번 구역으로 순간이동할 수 있음을 의미한다. (단, $u \neq v$)

출력

첫째 줄에 문제의 답을 출력한다.

제한

- $2 \leq N \leq 5,000$
- $1 \leq M \leq 50,000$
- $1 \leq T \leq 100$
- $1 \leq a_i \leq 10^9$
- $1 \leq x_i, y_i, c_i \leq 10,000$
- $1 \leq u, v \leq N, u \neq v$
- 입력으로 주어지는 모든 수는 양의 정수이다.

예제 입력 1

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

예제 입력 2

5 4 3 20 1 1 1 10 1 1 2 30 1 1 3 50 1 1 4 15 1 1 5 1 2 2 3 3 4 4 5	32
---	----

예제 입력 3

5 5 10 1 1 1 1 2 2 2 2 1 2 3 4 4 3 2 1 5 5 5 5 1 2 2 3 2 4 3 1 3 5	10
--	----