

2026 IUPC 문제지

문제 : 총 12 문항

문제지 : 총 24 장

※ 각 문제의 시간 및 메모리 제한은 해당 문제 상단에 명시되어 있습니다.

A. 5그와트	4 장
B. 흑흑요리사	1 장
C. BLACK or WHITE?	1 장
D. 도망치는 기계오리	2 장
E. IUPC 찾기	3 장
F. CTP 경로	2 장
G. 삼각형 만들기	1 장
H. 점프덕	2 장
I. max를 아끼는 법	2 장
J. 기억을 잃는 스택	1 장
K. 미생물	3 장
L. Artificial Ascent	1 장

※ 문제지와 사이트의 지문이 다르다면 사이트 지문을 우선으로 합니다.

A. 5그와트

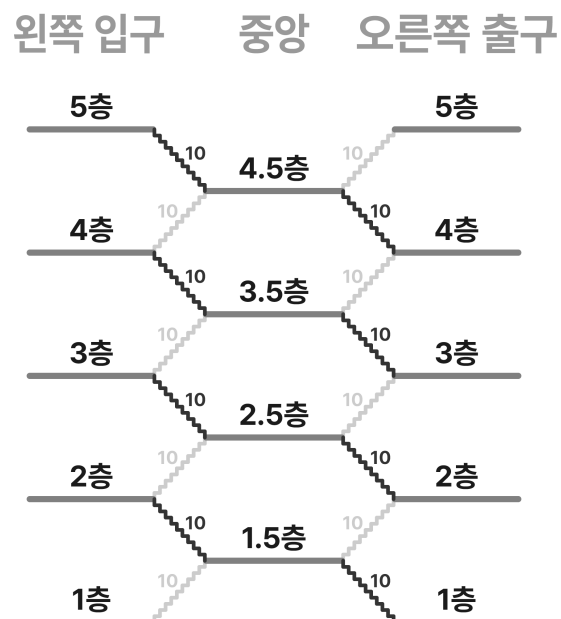
시간 제한 : 1초
메모리 제한 : 1024MB

문제



실제 5호관 남쪽 계단에 붙어 있는 안내문.

인하대학교 5호관 남쪽 계단은 독특한 구조 때문에 길을 헤매기 쉬워 '5그와트'라고 불린다. 계단의 구조는 다음 그림과 같다.



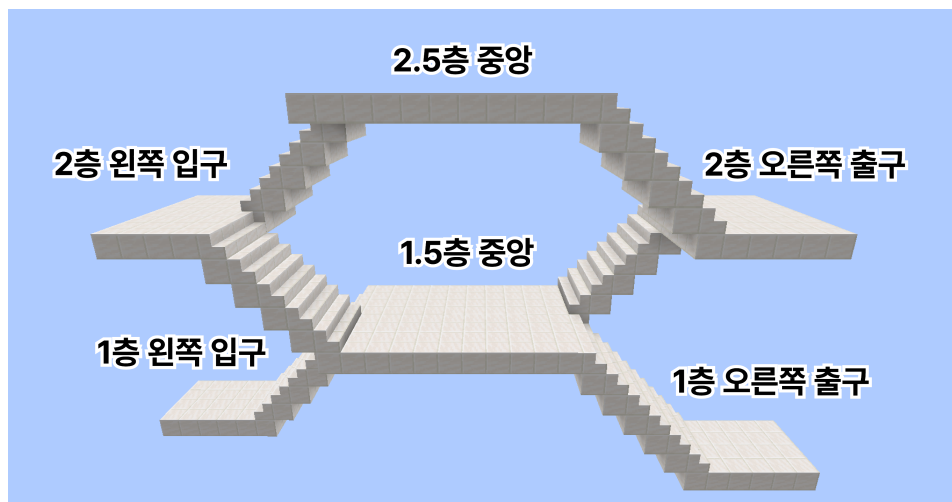
A. 5그와트

시간 제한 : 1초
메모리 제한 : 1024MB

5그와트에는 각 정수 층마다, 왼쪽에는 입구가 있고 오른쪽에는 출구가 있다. 이를 각각 왼쪽 입구, 오른쪽 출구라 한다. 정수 N 층 왼쪽 입구에서 정수 M 층 오른쪽 출구로 이동하려면, 반드시 계단 통로를 지나야 한다.

구체적으로, 5그와트의 계단 통로는 다음과 같은 규칙을 가진다. ($1 \leq K \leq 4$; K 는 정수)

- 문제에서 이용할 수 있는 통로는 아래에 설명한 계단뿐이다.
- 모든 계단은 양방향으로 이동할 수 있다. 즉, 0.5층을 올라가거나, 0.5층을 내려갈 수 있다.
- 0.5층을 올라가거나 내려갈 때 지나야 하는 계단의 총 칸 수는 항상 10칸이다.
- K 층 왼쪽 입구에서 $K + 0.5$ 층 중앙 지점으로 이어지는 계단이 존재한다.
- K 층 오른쪽 출구에서 $K + 0.5$ 층 중앙 지점으로 이어지는 계단이 존재한다.
- $K + 0.5$ 층 중앙 지점에서 $K + 1$ 층 왼쪽 입구로 이어지는 계단이 존재한다.
- $K + 0.5$ 층 중앙 지점에서 $K + 1$ 층 오른쪽 출구로 이어지는 계단이 존재한다.



5그와트에서 길을 헤매는 새내기들을 위해, 정수 N 층 왼쪽 입구에서 정수 M 층 오른쪽 출구까지 이동할 때 올라가거나 내려가야 하는 계단 칸 수의 최솟값을 구해주자!

입력

첫 번째 줄에 정수 N 과 M 이 공백으로 구분되어 주어진다. ($1 \leq N \leq 5, 1 \leq M \leq 5$)

출력

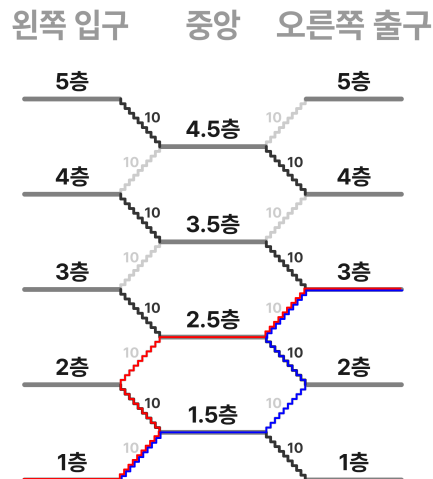
첫 번째 줄에 올라가거나 내려가야 하는 계단 칸 수의 최솟값을 출력한다.

A. 5그와트

시간 제한 : 1초
메모리 제한 : 1024MB

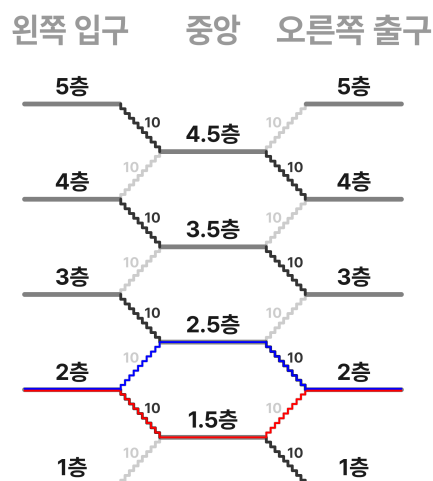
예제

입력 1	출력 1
1 3	40



첫 번째 예제에서는 다음 그림과 같이 두 가지 경로가 있고, 두 경로 모두 40칸의 계단을 올라가거나 내려가야 한다.

입력 2	출력 2
2 2	20

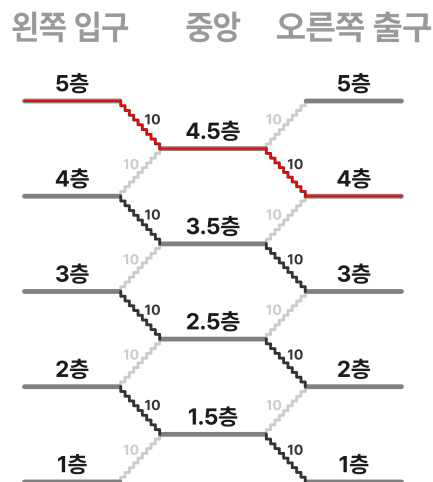


두 번째 예제에서는 다음 그림과 같이 두 가지 경로가 있고, 두 경로 모두 20칸의 계단을 올라가거나 내려가야 한다.

A. 5그와트

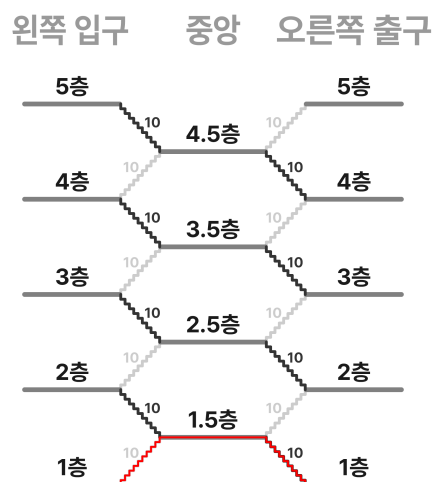
시간 제한 : 1초
메모리 제한 : 1024MB

입력 3	출력 3
5 4	20



세 번째 예제에서는 다음 그림과 같이 한 가지 경로가 있고, 20칸의 계단을 올라가거나 내려가야 한다.

입력 4	출력 4
1 1	20



네 번째 예제에서는 다음 그림과 같이 한 가지 경로가 있고, 20칸의 계단을 올라가거나 내려가야 한다.

B. 흑흑요리사

시간 제한 : 1초
메모리 제한 : 1024MB

문제

요리 경연 프로그램 흑흑요리사에 참가한 당신은 안용모 심사 위원에게 요리를 심사받아야 한다. 안 심사 위원은 요리의 익힘 정도와 간의 정확도를 매우 중요하게 생각한다. 구체적으로, 심사 기준은 다음과 같다.

- 요리의 익힘 정도를 A , 요리의 엄도를 B 라고 하자. $|(100 - A)(100 - B)| < 100$ 을 만족하면 당신은 이번 라운드를 통과할 수 있다.

지독한 구두쇠인 당신은 최소한의 소금을 사용하여 이번 라운드를 통과하려 한다. 현재 요리의 엄도 B 는 0이며, 소금을 한 번 첨가할 때마다 B 가 1 증가한다.

요리의 익힘 정도 A 가 주어질 때, 이번 라운드를 통과하기 위해 소금을 첨가해야 하는 최소 횟수를 구하자. 제한 시간 종료가 임박했기 때문에 요리를 더 익히는 것은 불가능하다.

입력

첫째 줄에 정수 A 가 주어진다. ($0 \leq A \leq 1'000$)

출력

이번 라운드를 통과하는 것이 가능하다면 소금의 최소 첨가 횟수를 출력한다. 소금을 첨가하지 않아도 라운드를 통과할 수 있다면 0을 출력한다.

이번 라운드를 통과하는 것이 불가능하다면 -1을 대신 출력한다.

예제

입력 1	출력 1
3	99

C. BLACK or WHITE?

시간 제한 : 0.5초
메모리 제한 : 1024MB

문제

치훈이를 포함한 N 명의 인하대학교 학생이 한 방에 모여 있으며, 각 학생은 검은색 모자 또는 흰색 모자 중 하나를 쓰고 있다.

전체 학생들에 대해 쓰고 있는 모자의 정보는 다음과 같다.

- 검은색 모자를 쓴 학생은 최소 A 명이다.
- 흰색 모자를 쓴 학생은 최소 B 명이다.

치훈이는 자신을 제외한 나머지 $N - 1$ 명의 학생들이 쓰고 있는 모자 색을 모두 볼 수 있으며, 그 결과

- 검은색 모자를 쓴 학생 C 명,
- 흰색 모자를 쓴 학생 D 명

을 보았다.

학생들 사이에는 어떠한 의사소통도 허용되지 않는다고 할 때, 치훈이가 항상 자신의 모자 색을 정확히 추론할 수 있는지 알아내 보자.

입력

한 줄에 다섯 정수 N, A, B, C, D 가 공백으로 구분되어 주어진다. 입력은 조건을 만족하는 상황이 적어도 하나 존재하도록 주어진다. ($1 \leq N \leq 100; 0 \leq A, B \leq N; 0 \leq C, D \leq N - 1$)

출력

치훈이가 자신의 모자 색이 검은색이라고 추론할 수 있다면 B, 흰색이라고 추론할 수 있다면 W, 추론할 수 없다면 X를 출력한다.

예제

입력 1	출력 1
5 2 2 1 3	B
입력 2	출력 2
5 3 1 4 0	W
입력 3	출력 3
5 1 1 2 2	X

D. 도망치는 기계오리

시간 제한 : 1초
메모리 제한 : 1024MB

문제

인하가 실험 장비를 만지다가 실수로 경보를 울렸고, 잠들어 있던 기계오리가 깨어나고 말했다! 경보음에 놀란 기계오리는 연구실을 뛰쳐나와 복도로 향했다.

인하는 기계오리를 무서워해서 연구실 밖으로 따라 나갈 수 없다. 대신 연구실 내부에서, 복도에 설치된 **도망 방지 시스템**을 원격으로 조작해 기계오리의 이동을 차단하려고 한다.

복도에는 도망 방지 시스템과 연동된 B 개의 자물쇠가 있으며, 각 자물쇠는 잠겨 있거나 열려 있다. 모든 자물쇠의 상태를 순서대로 나타낸 길이 B 의 이진 문자열을 자물쇠 상태라고 하자.

자물쇠 상태는 다음과 같이 표현된다.

- 문자열의 j 번째 문자가 1이면, j 번째 자물쇠는 잠겨 있다.
- 문자열의 j 번째 문자가 0이면, j 번째 자물쇠는 열려 있다.

초기 자물쇠 상태는 문자열 X , 목표 자물쇠 상태는 문자열 Y 로 주어진다.

도망 방지 시스템에는 1번부터 N 번까지 번호가 매겨진 N 개의 버튼이 있다. i 번 버튼에는 길이 B 의 이진 문자열 T_i 가 적혀 있다.

인하는 한 번에 하나의 버튼만 누를 수 있으며, 같은 버튼을 여러 번 누를 수도 있다. 인하가 i 번 버튼을 누르면, 모든 j ($1 \leq j \leq B$)에 대해 다음이 동시에 일어난다.

- T_i 의 j 번째 문자가 1이면, j 번째 자물쇠의 상태를 바꾼다. 즉, 열려 있으면 잠그고, 잠겨 있으면 연다.
- T_i 의 j 번째 문자가 0이면, 해당 자물쇠의 상태는 변하지 않는다.

시스템은 매우 예민하여, 자물쇠 상태가 Y 와 정확히 일치하는 순간 복도 전체에 셔터가 내려온다. 이때 기계오리는 도망칠 수 없게 된다.

인하는 버튼을 가능한 한 적게 눌러 자물쇠 상태를 Y 로 만들고 싶다. 인하를 도와, 자물쇠 상태를 Y 로 만들 수 있는지 판단하고, 가능하다면 필요한 버튼 누름 횟수의 최솟값을 구하자.

입력

첫째 줄에 B 와 N 이 공백으로 구분되어 주어진다. ($1 \leq B \leq 10; 1 \leq N \leq 10'000$)

둘째 줄에 길이 B 의 이진 문자열 X 가 주어진다.

셋째 줄에 길이 B 의 이진 문자열 Y 가 주어진다.

그다음 N 개의 줄 중 i 번째 줄에는 길이 B 의 이진 문자열 T_i 가 주어진다. ($1 \leq i \leq N$)

출력

자물쇠 상태를 Y 로 만들 수 있다면 필요한 버튼 누름 횟수의 최솟값을 출력한다.

자물쇠 상태를 Y 로 만들 수 없다면 -1을 출력한다.

D. 도망치는 기계오리

시간 제한 : 1초
메모리 제한 : 1024MB

예제

입력 1	출력 1
4 5 0000 1111 1000 0100 1100 0101 0010	3

입력 2	출력 2
3 3 000 100 010 001 011	-1

노트

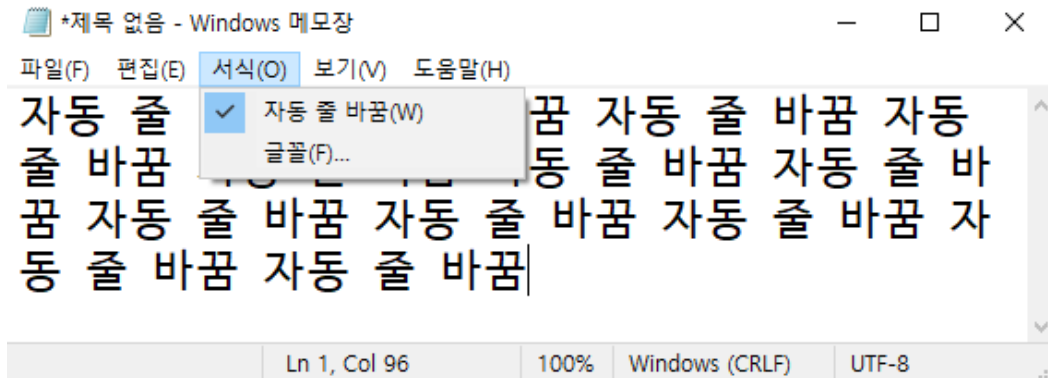
길이 L 의 이진 문자열이란 0과 1로만 이루어진 길이 L 의 문자열을 말한다.

E. IUPC 찾기

시간 제한 : 1초
메모리 제한 : 1024MB

문제

길이 N 의 문자열 S 가 주어진다. 메모장의 '자동 줄 바꿈' 기능을 켜고 문자열 S 를 무한히 이어 붙였을 때, 메모장의 너비를 적절히 조절해 세로로 된 'IUPC'를 찾을 수 있는지 판단하자.



문제 상황의 보다 엄밀한 정의는 다음과 같다.

길이 N 인 문자열 S 를 무한히 이어 붙여 만든 문자열을 T 라고 하고, T 의 i 번째 글자를 T_i ($1 \leq i$)라고 하자. 또, 행이 무한히 많고 열이 C 개인 격자 A 의 i 행 j 열에 적힌 문자를 $A_{i,j}$ ($1 \leq i, 1 \leq j \leq C$)라고 하자. 이때 $A_{i,j}$ 에 적힌 문자는 $T_{(i-1) \times C + j}$ 이다.

C 를 어떤 양의 정수로 정했을 때, 다음을 모두 만족하는 정수쌍 (i, j) 가 존재한다면 'IUPC'를 찾을 수 있다고 정의한다. ($1 \leq i, 1 \leq j \leq C$)

- $A_{i,j}$ 가 문자 'I'이다.
- $A_{i+1,j}$ 가 문자 'U'이다.
- $A_{i+2,j}$ 가 문자 'P'이다.
- $A_{i+3,j}$ 가 문자 'C'이다.

입력

첫 번째 줄에 문자열 S 의 길이를 나타내는 정수 N 이 주어진다. ($1 \leq N \leq 3'000$)

두 번째 줄에 알파벳 대문자로만 이루어진 길이 N 의 문자열 S 가 주어진다.

출력

세로로 된 'IUPC'를 찾을 수 있으면 YES, 찾을 수 없다면 NO를 출력한다.

E. IUPC 찾기

시간 제한 : 1초
메모리 제한 : 1024MB

예제

입력 1	출력 1
19 INHAUNIVPROGCONTEST	YES

입력: I N H A U N I V P R O G C O N T E S T

```

I N H A U N I V P R O G C O N T E S T I N H A
U N I V P R O G C O N T E S T I N H A U N I V
P R O G C O N T E S T I N H A U N I V P R O G
C O N T E S T I N H A U N I V P R O G C O N T
E S T I N H A U N I V P R O G C O N T E S T I
N H A U N I V P R O G C O N T E S T I N H A U
N I V P R O G C O N T E S T I N H A U N I V P

```

첫 번째 예제에서, 메모장의 너비 C 를 23으로 정하면 그림과 같이 'IUPC'를 찾을 수 있다.

입력 2	출력 2
28 CONTESTPROGRAMUNIVERSITYINHA	YES

입력: C O N T E S T P R O G R A M U N I V E R S I T Y I N H A

```

C O N T E S T P R O G R A M U N I V E R S
I T Y I N H A C O N T E S T P R O G R A M
U N I V E R S I T Y I N H A C O N T E S T
P R O G R A M U N I V E R S I T Y I N H A
C O N T E S T P R O G R A M U N I V E R S
I T Y I N H A C O N T E S T P R O G R A M
U N I V E R S I T Y I N H A C O N T E S T

```

두 번째 예제에서, 메모장의 너비 C 를 21로 정하면 그림과 같이 'IUPC'를 찾을 수 있다.

E. IUPC 찾기

시간 제한 : 1초
메모리 제한 : 1024MB

입력 3	출력 3
32 INHAUNIVERSITYPROGRAMMINGCONTEST	NO

세 번째 예제에서, 메모장의 너비 C 를 어떻게 정하더라도 'IUPC'를 찾을 수 없다.

노트

메모장에서 직접 확인할 때는 '굴림체'나 'Consolas'와 같은 고정폭 글꼴을 적용하고 '자동 줄 바꿈' 기능을 켜면 된다.

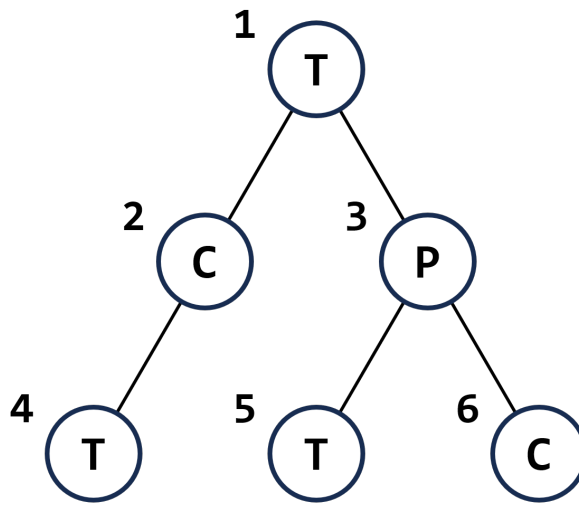
F. CTP 경로

시간 제한 : 5초
메모리 제한 : 1024MB

문제

1부터 N 까지 번호가 붙은 N 개의 정점으로 이루어진 트리가 있다. 각 정점에는 알파벳 C, T, P 중 하나가 쓰여 있다. 다음 조건을 만족하는 단순 경로를 CTP 경로라고 하자.

- 경로에 포함된 정점들의 문자를 순서대로 나열했을 때, 그 문자열이 "CTP"를 무한히 이어 붙인 "CTPCTPCT..." 문자열의 접두사이면 이 경로를 CTP 경로라고 한다.



예를 들어 위 그림의 트리에서 서로 다른 CTP 경로는 $[2], [2, 1], [2, 1, 3], [2, 1, 3, 6], [2, 4], [6]$ 이 있다.

당신은 CTP 경로가 최대한 많은 트리를 좋아한다. 트리에서 정점 하나를 골라서 적혀 있던 문자를 지우고 원하는 문자를 적어 넣어서 서로 다른 CTP 경로의 개수를 최대화해 보자. 지우기 전에 적혀 있던 문자와 같은 문자를 다시 적어 넣을 수도 있다.

입력

첫 번째 줄에 정수 N 이 주어진다. ($1 \leq N \leq 200'000$)

두 번째 줄에 1번 정점부터 N 번 정점까지 각 정점에 적혀 있는 문자들이 공백 없이 순서대로 주어진다. 각 문자는 'C', 'T', 'P' 중 하나이다.

세 번째 줄부터 $N - 1$ 개의 줄에 걸쳐서 간선의 정보가 주어진다. 간선은 u, v 의 형태로 주어진다. 이는 트리의 u 번 정점과 v 번 정점이 간선으로 연결되어 있음을 의미한다. ($1 \leq u, v \leq N$)

F. CTP 경로

시간 제한 : 5초
메모리 제한 : 1024MB

출력

첫 번째 줄에 수정할 정점 번호를 출력한다.

두 번째 줄에 적어 넣을 문자를 출력한다. 적어 넣을 문자는 'C', 'T', 'P' 중 하나다.

서로 다른 CTP 경로의 개수를 최대화하는 방법이 여러 가지라면, 그중 아무것이나 하나를 출력한다.

예제

입력 1	출력 1
6 TCPTTC 1 2 1 3 2 4 3 5 3 6	5 C

입력 2	출력 2
1 C	1 C

G. 삼각형 만들기

시간 제한 : 3초
메모리 제한 : 1024MB

문제

동현이는 길이가 1인 막대를 A 개, 길이가 2인 막대를 B 개, 길이가 3인 막대를 C 개, 길이가 4인 막대를 D 개 가지고 있다.

동현이는 이 막대들을 이용해 삼각형을 최대한 많이 만들려고 한다.

막대 3개를 선택하여, 그 길이들을 세 변의 길이로 하는 삼각형이 존재한다면 하나의 삼각형을 만들 수 있다. 즉, 가장 긴 막대의 길이가 나머지 두 막대 길이의 합보다 작다면 이 막대 3개로 삼각형 하나를 만들 수 있다. 단, 한 번 사용한 막대는 다시 사용할 수 없다.

동현이가 만들 수 있는 삼각형의 최대 개수를 구해보자.

입력

첫 번째 줄에 테스트 케이스의 개수 T 가 주어진다. ($1 \leq T \leq 100'000$)

각 테스트 케이스의 첫 번째 줄에 정수 A, B, C, D 가 주어진다. ($0 \leq A, B, C, D \leq 1'000$)

출력

각 테스트 케이스마다 동현이가 만들 수 있는 삼각형의 최대 개수를 한 줄에 하나씩 출력한다.

예제

입력 1	출력 1
3 1 1 1 4 4 0 1 1 0 0 0 0	2 1 0

H. 점프덕

시간 제한 : 1초
메모리 제한 : 1024MB

문제

인덕이는 '점프덕'이라는 게임을 개발했다. 안농이는 인덕이가 게임을 얼마나 잘 만들었는지 확인하기 위해 게임의 테스터가 되기로 했다.

점프덕의 게임판은 오른쪽으로 일렬로 이어진 N 개의 칸으로 이루어져 있다. 칸은 왼쪽부터 1번부터 N 번까지 순서대로 번호가 붙어 있다. 플레이어는 처음에 1번 칸에 있으며, N 번 칸이 목적지이다. 목적지를 제외한 i 번 칸 ($1 \leq i \leq N-1$)에는 음이 아닌 정수 A_i 가 적혀 있다. $A_i = 0$ 인 칸은 낭떠러지 칸이며, 플레이어는 이 칸으로 점프할 수 없다. $A_i > 0$ 인 칸에서는 1 이상 A_i 이하의 정수를 골라 그만큼 오른쪽으로 점프할 수 있다. 단, 점프한 뒤의 위치가 N 번 칸보다 오른쪽이 되는 점프는 할 수 없다. 목적지에 도착하면 게임을 클리어한다.

게임을 하다 화가 난 안농이는 게임 개발자 인덕이를 골탕 먹이기 위해, 1번 칸과 N 번 칸을 제외한 칸 몇 개를 낭떠러지 칸으로 바꾸어 게임을 클리어할 수 없게 만들려고 한다. 다만 너무 많은 칸을 낭떠러지 칸으로 바꿔버리면 인덕이가 눈치챌 것이기 때문에, 바꾸는 칸의 수를 최소화하려고 한다. 안농이를 위해 최소한의 칸만을 낭떠러지 칸으로 바꾸어 게임을 클리어할 수 없게 만드는 방법을 알려주자!

입력

첫 번째 줄에 칸의 수 N 이 주어진다. ($2 \leq N \leq 200'000$)

두 번째 줄에 $N-1$ 개의 정수 $A_1, A_2, \dots, A_{N-1}$ 이 공백으로 구분되어 주어진다. $A_i = 0$ 이면 i 번 칸이 처음부터 낭떠러지 칸임을 의미한다. ($1 \leq i \leq N-1, 0 \leq A_i \leq N-1$)

출력

게임을 클리어할 수 없게 만들기 위해 새로 낭떠러지 칸으로 바꿔야 하는 칸의 최소 개수를 K 라고 하자.

게임을 클리어할 수 없게 만드는 방법이 존재한다면 첫 번째 줄에 K 를 출력한다.

두 번째 줄에 새로 낭떠러지 칸으로 바꿀 서로 다른 칸 번호 K 개를 공백으로 구분하여 출력한다. 이때, 각 번호는 2 이상 $N-1$ 이하의 정수여야 한다. 출력 순서는 상관없다.

게임을 클리어할 수 없게 만드는 방법이 존재하지 않는다면 첫 번째 줄에 -1을 대신 출력한다.

H. 점프덱

시간 제한 : 1초
메모리 제한 : 1024MB

예제

입력 1	출력 1
10 3 1 4 1 5 9 2 6 5	2 3 5
입력 2	출력 2
4 2 1 0	0
입력 3	출력 3
5 4 1 2 1	-1

I. max를 아끼는 법

시간 제한 : 1초
메모리 제한 : 1024MB

문제

N 개의 양의 정수로 이루어진 두 수열 A, B 가 주어진다. 수열 A 에 아래 연산을 0회 이상 수행하여 수열 B 를 만들려고 한다.

1. $1 \leq l \leq r \leq N$ 인 두 정수 l, r 를 고른다.
2. 연산 직전의 수열 A 에서 $A_l, A_{l+1}, \dots, A_r$ 중 최댓값을 M 이라고 하자.
3. 모든 $l \leq i \leq r$ 에 대해 A_i 를 M 으로 바꾼다.

A 와 B 가 주어졌을 때, A 를 B 로 만들기 위해 필요한 연산의 최소 횟수를 구해보자.

입력

첫째 줄에 수열의 길이 N 이 주어진다. ($1 \leq N \leq 5'000$)

둘째 줄에 수열 A 의 원소 $A_1, A_2, \dots, A_N$ 이 공백으로 구분되어 주어진다. ($1 \leq A_i \leq N$)

셋째 줄에 수열 B 의 원소 $B_1, B_2, \dots, B_N$ 이 공백으로 구분되어 주어진다. ($1 \leq B_i \leq N$)

출력

A 를 B 로 만드는 데 필요한 연산의 최소 횟수를 출력한다.

A 를 B 로 만들 수 없다면 -1을 출력한다.

I. max를 아끼는 법

시간 제한 : 1초
메모리 제한 : 1024MB

예제

입력 1	출력 1
6 3 1 3 4 5 6 3 4 4 5 6 6	3
입력 2	출력 2
2 1 1 1 1	0
입력 3	출력 3
4 4 3 1 2 4 3 1 3	-1
입력 4	출력 4
4 4 3 1 2 4 1 1 2	-1
입력 5	출력 5
4 4 3 1 2 3 4 4 4	-1

J. 기억을 잃는 스택

시간 제한 : 1초
메모리 제한 : 1024MB

문제

다음 연산을 지원하는 스택을 구현해 보자. 초기 시각은 0이며, 시간은 `add_time` 연산으로만 흐른다.

- `push x f` : 정수 x 를 스택의 맨 위에 넣는다. 이 스택은 기억을 잃는 성질이 있다. 현재 시각이 T 라면, 이 연산으로 넣은 원소는 현재 시각이 $T + f$ 이상이 되는 순간 스택에서 사라진다.
($1 \leq x \leq 1'000'000$; $1 \leq f \leq 1'000'000$; f 는 정수)
- `pop` : 현재 시각에 스택에 남아 있는 원소 중 맨 위의 원소를 제거하고 그 값을 출력한다. 현재 시각에 스택에 남아 있는 원소가 없다면 -1을 출력한다.
- `add_time t` : 현재 시각이 t 만큼 증가한다. 현재 시각에 이미 사라져야 하는 원소들은 `pop` 연산을 수행하기 전에 모두 사라진다. ($1 \leq t \leq 1'000'000$; t 는 정수)

초기 시각에 스택은 비어 있다. N 회의 연산이 주어졌을 때, `pop` 연산이 주어질 때마다 제거되는 정수 또는 -1을 출력하자.

입력

첫 번째 줄에는 연산의 횟수 N 이 주어진다. ($1 \leq N \leq 1'000'000$)

이어서 N 개의 줄에 걸쳐, 연산이 한 줄에 하나씩 주어진다.

출력

`pop` 연산이 주어질 때마다 제거되는 정수 또는 -1을 한 줄에 하나씩 출력한다.

예제

입력 1	출력 1
8 push 2 3 push 5 5 push 7 1 add_time 1 pop add_time 1 pop pop	5 2 -1

노트

오버플로우에 조심하자.

K. 미생물

시간 제한 : 1초
메모리 제한 : 1024MB

문제

인경호 바닥에는 많은 미생물들이 살고 있다. 인경호 탐사를 마친 양코는 이곳의 미생물이 모두 완벽한 원판 모양이며, 두께가 무시할 수 있을 정도로 얇다는 사실을 알아냈다.

양코는 미생물들의 위치 관계를 다음과 같이 정의했다.

- 두 미생물이 서로 접하거나 겹쳐 있으면 이 두 미생물이 **직접 연결되어 있다**고 한다.
- 직접 연결 관계를 따라 서로 도달할 수 있는 미생물들의 집합을 하나의 **군체**라고 한다.

또한 양코는 이 미생물들이 다음 규칙에 따라 움직인다는 사실을 확인하였다.

1. 한 번의 회전 운동에서는 미생물 하나 m 을 회전 중심으로 선택하고, 회전 방향과 회전 각도를 임의로 정한다. 둘 이상의 미생물이 동시에 회전 중심으로 선택되는 경우는 없다.
2. 이때 m 이 속한 군체의 모든 미생물은 m 의 중심을 기준으로 하나의 강체처럼, 같은 방향으로 같은 각도만큼 회전한다.
3. 회전 중의 접촉은 고려하지 않는다. 회전이 끝난 뒤, 서로 다른 군체에 속한 미생물이 새로 접하거나 겹치면, 그렇게 연결된 군체들은 모두 하나의 군체로 합쳐진다.

이 미생물들은 추위에 약해, 겨울이 되기 전에 하나의 군체를 이뤄야 에너지를 공유하며 살아남을 수 있다.

인경호 바닥의 미생물들의 배치 정보가 주어질 때, 회전 운동을 유한 번 수행하여 모든 미생물들이 하나의 군체를 이룰 수 있는지 확인하는 프로그램을 작성해보자.

입력

첫 번째 줄에 미생물의 수 N 이 주어진다. ($2 \leq N \leq 5'000$)

이어서 N 개의 줄에 걸쳐 세 정수 x_i, y_i, r_i 가 공백으로 구분되어 주어진다. 이는 i 번째 미생물의 중심 좌표가 (x_i, y_i) 이고 반지름의 길이가 r_i 임을 의미한다. ($-10'000 \leq x_i, y_i \leq 10'000, 1 \leq r_i \leq 10'000$)

출력

회전 운동을 유한 번 수행하여 모든 미생물이 하나의 군체를 이룰 수 있으면 1, 그렇지 않으면 0을 출력한다.

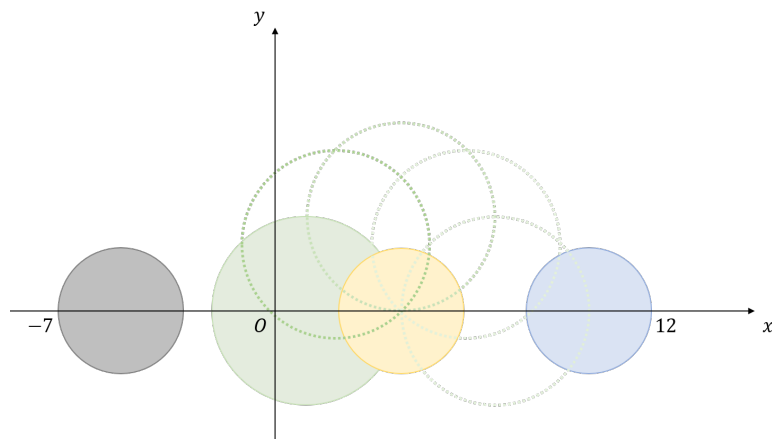
K. 미생물

시간 제한 : 1초
메모리 제한 : 1024MB

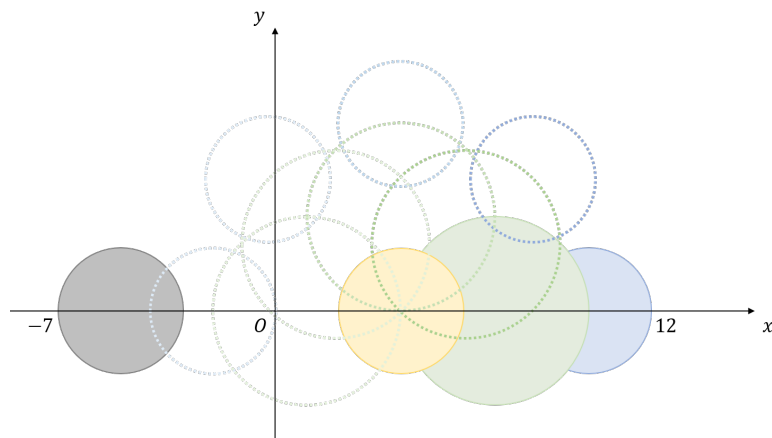
예제

입력 1	출력 1
4 -5 0 2 1 0 3 4 0 2 10 0 2	1

아래는 예제 1의 미생물들이 하나의 군체로 합쳐지는 방법 중 하나이다.



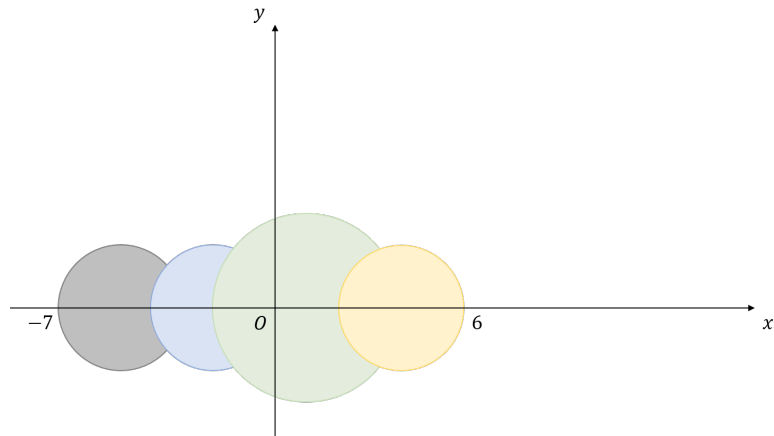
먼저, 3번 미생물이 시계 방향으로 180° 회전 운동하면 2, 3, 4번 미생물들이 하나의 군체를 이룬다.



이후 3번 미생물이 반시계 방향으로 180° 회전 운동하면 다음 페이지의 그림과 같이 모든 미생물들이 하나의 군체를 이루게 된다.

K. 미생물

시간 제한 : 1초
메모리 제한 : 1024MB



입력 2	출력 2
2 51 3 53 83 58 3	0

L. Artificial Ascent

시간 제한 : 1초
메모리 제한 : 1024MB

문제

길이가 N 이고 각 원소가 1 이상 9 이하의 양의 정수인 배열 P 에 대해 아래의 연산을 수행하려고 한다.

- 배열 P 의 원소를 임의로 재배열한 뒤, 재배열된 배열을 한 지점에서 나누어 앞쪽 원소들로 배열 X 를, 뒤쪽 원소들로 배열 Y 를 만든다. 두 배열 X 와 Y 는 모두 비어 있지 않아야 한다.
- 배열 X 의 원소를 순서대로 이어 붙여 양의 정수 A 를 만들고, 배열 Y 의 원소를 순서대로 이어 붙여 양의 정수 B 를 만든다.^[1]

이때 가능한 $A \bmod B$ 의 최댓값을 구해보자.^[2]

입력

첫 번째 줄에 배열 P 의 크기 N 이 주어진다. ($2 \leq N \leq 500'000$)

두 번째 줄에 $P_1, P_2, \dots, P_N$ 이 공백으로 구분되어 주어진다. ($1 \leq P_i \leq 9$)

출력

첫 번째 줄에 가능한 $A \bmod B$ 의 최댓값을 출력한다.

예제

입력 1	출력 1
2 3 5	3

입력 2	출력 2
3 3 1 6	6

노트

^[1] 길이가 n 의 배열 $C_1, C_2, \dots, C_n$ 을 순서대로 이어 붙인 양의 정수 K 의 수학적 정의는 아래와 같다.

$$K = \sum_{i=1}^n C_i \times 10^{n-i}$$

^[2] 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.