

2025 SCCC Contest - Div.1

Official Problem Set

Russia Team High School Programming Contest

2005-2010

Problem List

#	Problem Name	Time limit	Memory limit	Page
A	제비뽑기	2 seconds	1024MiB	3 – 3
B	숫자 놀이	2 seconds	1024MiB	4 – 4
C	영화관	2 seconds	1024MiB	5 – 5
D	생일 축하 퍼즐	2 seconds	1024MiB	6 – 7
E	다과제	2 seconds	1024MiB	8 – 8
F	최대합	2 seconds	1024MiB	9 – 10
G	자료구조 및 실습	2 seconds	1024MiB	11 – 12
H	서버 롤백	2 seconds	512MiB	13 – 13

모든 문제는 C++17로 풀 수 있음을 보장합니다.

A. 제비뽑기

창민이는 친구들과 때때로 제비뽑기 게임을 한다. 게임 규칙은 다음과 같다: 매 라운드마다 각 플레이어는 임의의 자연수를 하나씩 선택한다. 그 후, 선택된 수들 중에서 가장 작은 수이면서 중복되지 않는 수를 선택한 플레이어가 그 라운드의 승자가 되고, 그의 획득 점수는 그 수와 같다. 예를 들어 6명이 게임에 참여했고 선택된 수들이 3, 2, 1, 1, 4, 2라면 첫 번째 플레이어가 이기며 그의 획득 점수는 3이다. 만약 모든 선택된 수가 중복되어 단 하나의 수도 유일하지 않다면 그 라운드는 무승부로 처리되며 아무도 점수를 얻지 못한다.

게임에서의 한 플레이어의 총득점은 모든 라운드에서 얻은 점수의 합이다.

창민이와 친구들은 게임에서 보통 차례대로 자신들이 선택한 수를 말한 후 누가 이겼는지 정하고 점수를 합산한다. 그러나 이런 형식에서는 미리 수를 정해두지 않고 앞서 말한 플레이어들의 수를 들은 뒤에 최적의 수를 말함으로써 부정행위를 할 수 있다. 창민이가 바로 이 방법을 사용한다. 그는 항상 마지막에 수를 말하며 자신의 총득점을 최대화하도록 수를 선택하려 한다.

마지막 라운드가 진행 중이다. 마지막 라운드 직전의 모든 플레이어의 점수와, 마지막 라운드에서 앞선 플레이어들이 이미 말한 수들이 주어져 있다. 창민이가 어떤 수를 말해야 라운드 결과 후 가능한 많은 플레이어들보다 그의 총점이 더 크도록 할 수 있는지 구하시오. 그러한 수가 여러 개인 경우 창민이는 가능한 한 가장 작은 수를 말하고 싶어 한다. 창민이는 n 번째 플레이어이다.

입력 형식

첫 번째 줄에는 플레이어의 수 n 이 주어진다. ($2 \leq n \leq 100$).

두 번째 줄에는 마지막 라운드 직전의 플레이어들의 점수 a_i 가 n 개 주어진다. ($0 \leq a_i \leq 100$)

세 번째 줄에는 마지막 라운드에서 플레이어들이 선택한 $n - 1$ 개의 수 b_i 가 주어진다. ($b_i \leq 100$)

출력 형식

창민이가 말해야 할 수 하나를 출력하시오.

예제

표준 입력(stdin)	표준 출력(stdout)
6 0 0 0 0 0 0 2 3 4 5 6	1
6 8 3 12 5 0 9 2 1 3 1 4	2

B. 숫자 놀이

아이들은 마당에서 술래잡기를 하기로 했다. 술래를 정하기 위해, 그들은 숫자 놀이(считалка)를 이용하기로 했다. 숫자 놀이는 k 개의 단어로 이루어져 있으며, 다음과 같이 사용된다.

모든 n 명의 아이들이 원을 이루어 서고, 한 아이가 자기 자신부터 시작하여 원을 따라 한 방향으로 옆에 있는 다른 친구를 가리키며 숫자 놀이의 단어들을 말한다. 마지막 k 번째 단어를 말할 때 가리켜진 아이가 원에서 제외된다. 그 다음 숫자 놀이는 처음부터 다시 시작되며, 방금 제외된 아이의 다음 순서의 아이부터 단어 말하기를 계속한다. 이 과정은 원에 한 명만 남을 때까지 반복되며, 그 마지막 남은 아이가 술래가 된다.

그러나 아이들이 다가올 게임에 너무 몰두한 나머지, 숫자 놀이에서 마지막 단어를 불러 제외되어야 할 때도 원을 떠나는 것을 잊어버렸다. 그 결과, 어떤 아이도 이후의 숫자 놀이에서도 여전히 같은 아이를 가리켰다.

아이들은, 숫자 놀이의 한 반복이 끝났을 때, 처음 시작한 아이가 마지막 단어에서 이미 원을 떠났어야 할 아이를 또 다시 가리킨 순간에야 아무도 떠나지 않았다는 사실을 깨달았다. 이제 그들은 이 시점에서 아직까지 ‘원에서 나가라’고 가리킨 적 없는 아이가 몇 명 남아 있는지 궁금해졌다.

이들에게 답을 구해주자.

입력 형식

입력 파일에는 두 개의 정수 — n 과 k 가 주어진다. ($1 \leq n \leq 1000$, $1 \leq k \leq 10^9$)

출력 형식

처음 시작한 아이가 마지막 단어에서 어떤 아이를 다시 가리키는 순간, 아직 ‘원에서 나가라’고 가리킨 적 없는 아이들의 수를 출력한다.

예제

표준 입력(stdin)	표준 출력(stdout)
6 14	3
6 13	0

C. 영화관

창민이는 문성이와 함께 SCCC 부원들을 영화관에 데리고 왔다. 불만이 생기지 않도록, 창민이는 모든 부원들을 알파벳 순으로 세운 뒤 자리에 앉혔다: 먼저 첫 번째 줄에 왼쪽에서 오른쪽으로, 그다음 두 번째 줄에 왼쪽에서 오른쪽으로, 이런 식으로 n 줄에 m 개의 좌석을 전부 채웠다.

그런데 문성이가 와서 부원들이 잘못 앉았다고 하며 다시 앉으라고 했다. 그는 먼저 각 줄의 첫 번째 좌석들을 위에서부터 아래로 채우고, 그다음 두 번째 좌석들을 채우고, 이런 식으로 하자고 제안했다.

이때, 이런 재배치 후에도 자기 자리에 그대로 남아 있는 학생이 몇 명인지 구하시오.

예를 들어, $n = 3$, $m = 3$ 이라면 첫 번째 경우(행 우선 배치)에는 학생들이 다음과 같이 앉는다:

```
1 2 3
4 5 6
7 8 9
```

두 번째 경우(열 우선 배치)에는 다음과 같다:

```
1 4 7
2 5 8
3 6 9
```

따라서 학생 1, 5, 9 — 세 명이 자기 자리 그대로 남는다.

입력 형식

입력 파일에는 두 정수 n 과 m 이 주어진다. ($1 \leq n, m \leq 10^9$).

출력 형식

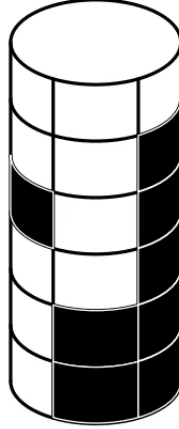
자기 자리에 남아 있는 학생의 수를 출력하시오.

예제

표준 입력(stdin)	표준 출력(stdout)
3 3	3
2 4	2

D. 생일 축하 퍼즐

은규는 생일을 맞아 새로운 퍼즐을 받았다. 이 퍼즐은 세로 축에 꿰어진 n 개의 원형 층으로 이루어진 원통 모양이다. 각 층은 다른 층과 독립적으로 회전할 수 있다. 각 층은 n 개의 네모 칸으로 나누어져 있으며, 각 칸은 검은색 또는 흰색이다.



퍼즐의 한 상태를 표현하기 위해, 원통의 옆면을 세로 칸의 경계선을 따라 절단 후 펼친 전개도를 생각해 보자. 검은 칸은 1, 흰 칸은 0로 표시한다. 위 그림에서 나타난 퍼즐의 전개도는 다음과 같다.

```
000110
001110
101000
001000
011111
011110
```

퍼즐의 목표는 각 층을 회전시켜, 전개도의 모든 세로 열이 서로 다르게 만드는 것이다. 위의 예시는 아직 풀린 것이 아닌데, 그 이유는 전개도의 4번째 열과 5번째 열이 동일하기 때문이다. 하지만, 가장 아래 층을 왼쪽으로 한 칸만큼 회전시키면, 전개도는 다음과 같이 바뀐다.

```
000110
001110
101000
001000
011111
111100
```

이제 모든 열이 서로 달라졌으므로, 퍼즐은 해결된 상태가 된다.

퍼즐을 더 흥미롭게 만들기 위해, 은규는 각 층의 무늬에 다음과 같은 조건을 추가했다.

한 층을 완전히 한 바퀴 회전하기 전까지는 같은 모양으로 돌아오지 않는다.

예를 들어, $n = 6$ 일 때, 층의 무늬가 010101인 경우는 허용되지 않는다. 왜냐하면 이 무늬는 층을 2칸 돌렸을 때 원래 모양으로 돌아오기 때문이다. 주어진 퍼즐을 풀어 은규의 생일을 축하해주자.

입력 형식

첫 번째 줄에 퍼즐의 행과 열의 크기를 나타내는 정수 n 이 주어진다. ($1 \leq n \leq 200$)

다음 n 개의 줄에 퍼즐의 전개도를 나타내는 n 개의 문자가 주어진다. 각 문자는 0 또는 1로 이루어져 있다.

출력 형식

퍼즐을 풀 수 없다면, 첫 줄에 No를 출력한다.

퍼즐을 풀 수 있다면, 첫 줄에 Yes를 출력한 후, 다음 n 개의 줄에 걸쳐, 퍼즐이 풀린 전개도를 출력한다.

가능한 답이 여러가지인 경우, 그 중 아무거나 하나 출력한다.

예제

표준 입력(stdin)	표준 출력(stdout)
6	Yes
000110	000110
001110	011100
101000	101000
001000	001000
011111	011111
011110	011110

E. 다과제

은규는 종종 과제를 여러 PC로 진행하곤 한다. 은규가 가진 PC는 두 종류로, 데스크탑과 노트북이다. 각 PC에는 하나의 과제가 미리 할당되어 있으며 어떤 과제들은 특정한 선행 과제들을 풀어야만 진행할 수 있다.

매번 PC의 종류를 바꿔 가면서 과제를 풀면 굉장히 번거롭기 때문에, 은규는 PC의 종류를 바꾸는 횟수를 최소화하면서 모든 과제를 풀고자 한다. 은규를 위해 어떤 순서로 과제를 풀어야 하는지 알려주자.

입력 형식

첫째 줄에는 과제의 수 n 이 주어진다 ($1 \leq n \leq 100$).

다음 n 개의 줄에는 각 과제들의 정보가 주어진다. 각 과제들은 어떤 임의의 순서로 1부터 n 까지 번호가 매겨져 있다고 하자. 그러면 i 번째 줄은 i 번째 과제를 설명한다.

각 과제는 다음과 같은 형태로 주어진다.

1. 첫 번째 정수는 과제가 할당된 PC의 종류이며, 0은 데스크탑, 1은 노트북을 의미한다.
2. 그 다음 정수 r_i 는 이 과제를 풀기 위해 먼저 완료되어야 하는 과제들의 수이다.
3. 그 뒤에는 r_i 개의 정수가 주어지며, 이는 반드시 먼저 수행해야 하는 과제들의 번호를 의미한다. 이들은 각각 1부터 $i - 1$ 이하의 서로 다른 정수이다.

모든 과제를 해결 가능한 케이스만 주어진다.

출력 형식

첫째 줄에는 은규가 PC의 종류를 바꾸는 최소 횟수를 출력한다.

둘째 줄에는 은규가 과제를 해결하는 1부터 n 까지의 수의 한 순열을 출력한다.

최적해가 여러 개라면 그중 아무 것이나 출력하면 된다.

예제

표준 입력(stdin)	표준 출력(stdout)
3	1
1 0	2 1 3
0 0	
1 2 1 2	

F. 최대 합

오늘 신문 "수학 오락" 지면에는 특별한 수학 퍼즐이 실렸다. 신문의 한 페이지 전체가 m 행과 n 열로 이루어진 직사각형 표로 채워져 있으며, 표의 각 칸에는 어떤 정수가 적혀 있다.

퍼즐의 목표는 표의 칸 중심을 꼭짓점으로 하고, 변이 표의 변과 평행한 한변의 길이가 2이상인 직사각형을 선택하는 것이다. 그리고 그 직사각형의 경계에 위치한 칸들에 적힌 수들의 합이 최대가 되도록 찾아야 한다.

몇 시간을 허비하고도 퍼즐을 풀지 못한 민성이는 이를 대신해 줄 프로그램을 작성하려 했지만, 결국 실패했다. 이제 그는 여러분에게 도움을 요청한다.

주어진 표에 대해, 조건을 만족하는 직사각형을 찾아라.

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

입력 형식

첫 번째 줄에는 두 정수 m 과 n 이 주어진다. $2 \leq m, n \leq 300$ 이후 m 개의 줄이 주어지며 각 줄에는 n 개의 정수 a_{ij} 가 주어진다. ($-10^4 \leq a_{ij} \leq 10^4$)

출력 형식

첫 번째 줄에 찾은 직사각형의 경계 칸들에 적힌 수들의 최대 합을 하나의 정수로 출력한다.

두 번째 줄에는 네 개의 자연수 x_1, y_1, x_2, y_2 각각 선택한 직사각형의 왼쪽위, 오른쪽 아래칸 좌표를 출력한다.

최적해가 여러개라면 그중 아무거나 출력해도 된다.

예제

표준 입력(stdin)	표준 출력(stdout)
2 3 1 1 1 1 1 1	6 1 1 2 3
5 4 9 -2 -1 3 -10 -5 1 -4 1 -1 2 -2 3 0 0 -1 2 2 -1 2	8 3 1 5 3

G. 자료구조 및 실습

은규는 이번 학기 글로벌미디어학부에서 개설된 자료구조 및 실습 강의 시간에 레드-블랙 트리를 구현할 예정이다. 레드-블랙 트리의 정의는 다음과 같다.

레드-블랙 트리는 이진 트리로 구성되어 있다. 이진 트리는 각각의 노드가 최대 두개의 자식 정점을 가지는 트리를 말하며, 어떤 이진 트리가 레드-블랙 트리라고 불리려면, 각 정점이 빨강 혹은 검정으로 색칠되어 있어야 하며, 동시에 다음 조건들을 만족해야 한다.

1. 만약 정점이 빨강이라면, 그 부모 정점은 검정이어야 한다.
2. 루트에서 적어도 한쪽 자식이 없는 임의의 정점까지 이르는 경로에 있는 검정 정점의 개수는 모두 동일해야 한다.

색칠된 정점을 검정, 색칠되지 않은 정점을 빨강이라고 하자. 아래는 두 가지 색으로 칠해진 이진 트리의 예시다.

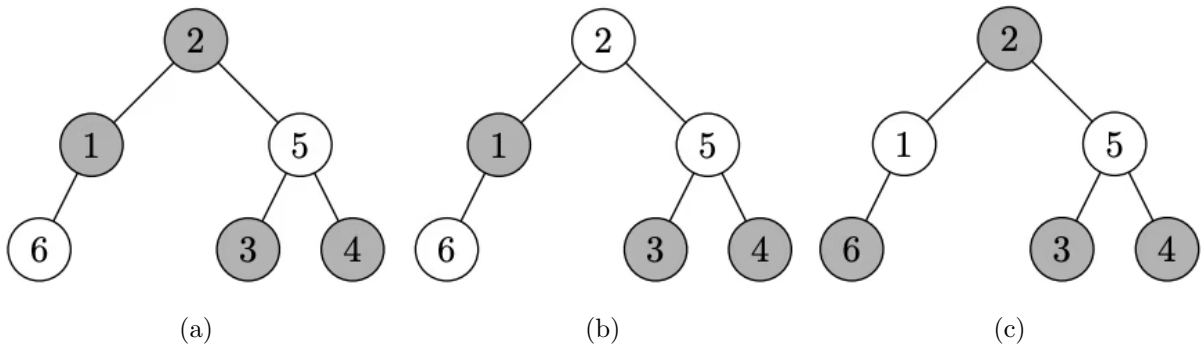


그림 (a)의 트리는 레드-블랙 트리다.

그림 (b)에서는 빨강 정점 5의 부모 정점 2 역시 빨강이므로 레드-블랙 트리가 아니다.

그림 (c)에서는 루트에서 정점 1까지의 경로에는 검정 정점이 1개인데, 루트에서 정점 3까지의 경로에는 검정 정점이 2개이므로 레드-블랙 트리가 아니다.

은규는 문득 이진 트리가 주어졌을 때, 이 트리를 색칠하여 레드-블랙 트리로 만드는 색칠의 수가 궁금해졌다. 은규를 위해 이를 구해주자.

입력 형식

첫 번째 줄에는 트리의 정점의 개수 자연수 n 이 주어진다. ($1 \leq n \leq 1000$)

다음 n 개의 줄에 걸쳐 자식 정점의 정보가 주어진다. $i+1$ 번째 줄에는 정점 i 의 왼쪽 자식과 오른쪽 자식의 정점 번호가 순서대로 주어진다. 만약 자식이 없다면 0이 주어진다.

출력 형식

주어진 이진 트리에 대해 가능한 레드-블랙 트리의 개수를 출력한다. 단, 숫자가 매우 커질 수 있으므로 $10^9 + 7$ 로 나눈 나머지를 출력한다.

예제

표준 입력(stdin)	표준 출력(stdout)
6 6 0 1 5 0 0 0 0 3 4 0 0	3
4 2 0 3 0 4 0 0 0	0

H. 서버 롤백

창민이는 아주 큰 회사에서 시스템 관리자로 일하고 있다. 당연히, 그의 업무 범위에는 여러 서버에 저장된 정보의 백업과, 문제가 발생했을 경우 이전 버전으로의 **롤백**하는 일이 포함된다.

현재 창민이는 롤백을 위한 저장 공간이 부족한 문제를 해결하고 있다. 고민 끝에 일부 정보를 새로운 서버로 옮기기로 결정했지만, 전송 도중 문제가 발생하면 롤백을 수행할 수 없으므로, 데이터 전송 과정은 신중하게 계획해야 한다.

현재 창민이는 서로 다른 서버에 대한 n 개의 복구 지점을 가지고 있으며, 각 복원 지점은 1번부터 n 번까지 번호가 매겨져 있다. i 번 복구 지점은 서버 a_i 에 대한 롤백을 지원한다.

창민이는 데이터 전송을 여러 단계로 나누고, 각 단계에서 문제가 발생했을 때 이용 가능한 복원 지점의 범위가 $[l, r]$ 로 주어진다고 가정한다. 즉, 어떤 l 과 r 에 대해 복구 지점 번호 $l, l+1, \dots, r-1, r$ 가 사용 가능하다.

데이터 전송을 최적으로 계획하기 위해서는 다음과 같은 질의에 답해야 한다.

- 주어진 l 에 대해서, 최소한 k 개의 서로 다른 서버에 대한 복원 지점을 포함하는 가장 작은 r 은 무엇인가?

입력 형식

첫 번째 줄에는 입력 파일의 첫 번째 줄에는 복구 지점의 개수 정수 n 과 서버의 개수 정수 m 이 공백으로 구분되어 주어진다. ($1 \leq n, m \leq 10^5$)

두 번째 줄에는 n 개의 정수 a_1, a_2, \dots, a_n 이 공백으로 구분되어 주어진다. 여기서 a_i 는 i 번째 복구 지점에 대응되는 서버의 번호이다. ($1 \leq a_i \leq m$)

입력 파일의 세 번째 줄에는 쿼리의 개수 정수 q 가 주어진다. ($1 \leq q \leq 10^5$).

다음 q 개의 줄에 걸쳐 쿼리가 주어진다. 질의를 처리하는 과정에서 수 p 를 유지해야 하며, 처음에는 $p = 0$ 이다. 각 쿼리는 두 개의 수 x_i, y_i 로 주어지며, 이 수들을 사용하여 쿼리의 데이터를 다음과 같이 얻는다.

$$l_i = ((x_i + p) \bmod n) + 1, k_i = ((y_i + p) \bmod m) + 1, (1 \leq l_i, x_i \leq n), (1 \leq k_i, y_i \leq m)$$

i 번째 질의의 답을 r 이라 하자. 이 질의가 처리된 후에는 $p := r$ 로 값을 갱신한다.

출력 형식

각 쿼리에 대해, 답 r 을 한 줄에 출력한다. 만약 그러한 r 이 존재하지 않는다면, 대신 0을 출력한다.

예제

표준 입력(stdin)	표준 출력(stdout)
7 3	1
1 2 1 3 1 2 1	4
4	0
7 3	6
7 1	
7 1	
2 2	