

# **2025 SCCC Contest - Div.2**

## Official Problem Set

# Russia Team High School Programming Contest

## 2005-2010

### Problem List

#	Problem Name	Time limit	Memory limit	Page
A	기사의 방패	2 seconds	512MiB	3 – 3
B	제비뽑기	2 seconds	1024MiB	4 – 4
C	숫자 놀이	2 seconds	1024MiB	5 – 5
D	공정한 수열	2 seconds	1024MiB	6 – 6
E	측정기 놀이	2 seconds	1024MiB	7 – 7
F	AREA 998244353	2 seconds	1024MiB	8 – 8
G	다과제	2 seconds	1024MiB	9 – 9
H	최대합	2 seconds	1024MiB	10 – 11

모든 문제는 C++17, Pypy3, Java로 풀 수 있음을 보장합니다.

## A. 기사의 방패

기사에게는 용과의 전투를 위해 갑옷, 방패, 그리고 검이 필요했다. 누구나 알다시피 방패는 클수록 전투에서 더 효과적이다. 지금 기사에게는 삼각형 모양의 방패 두 개가 있는데, 합쳐서 하나의 방패를 만들고자 했다.

왕국의 대장장이가 방패 제작을 맡았는데, 그는 다음과 같은 방법을 제안했다: 두 개의 방패를 서로 나란히 놓아 한 변이 맞닿게 한 다음, 그 상태로 고정시키는 것이다. 기사는 대장장이가 어떻게 고정시키든, 결과적으로 얻어지는 방패의 넓이는 항상 같을 것이며, 따라서 용과의 전투에서 방패의 효율은 대장장이가 어떻게 방패를 붙였는지가 아니라, 어떤 방패를 주었는지에만 달려 있다는 점을 알아차렸다.

하지만 그에게 필요한 것은 단순한 금속 조각이 아니라, 가문의 상징을 지닌 방패였다. 즉, 둘레 전체에 금으로 장식을 두른 방패였다. 그런데 지금 금값이 비싸므로, 기사는 결과적으로 얻어지는 방패의 둘레가 가능한 한 최소가 되기를 원했다. 그가 만들 수 있는 방패의 최소 둘레가 얼마인지 알아내는 것을 도와주자.

### 입력 형식

첫 번째 줄에는 첫 번째 방패의 세 변의 길이  $a_1, b_1, c_1$  이 주어진다.

두 번째 줄에는 두 번째 방패의 세 변의 길이  $a_2, b_2, c_2$  가 주어진다.

두 삼각형 모두 모두 가장 큰 변의 길이가 나머지 변의 길이의 합보다 작은 올바른 삼각형이다.

모든 수는 100,000 을 넘지 않는다.

### 출력 형식

조건에 맞게 제작할 수 있는 방패의 최소 둘레를 하나의 수로 출력한다.

### 예제

표준 입력(stdin)	표준 출력(stdout)
1 1 1 1 1 1	4
3 4 5 8 7 6	23

## B. 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

## C. 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

## D. 공정한 수열

짝수 길이의 0과 1로 이루어진 수열 중, 그 수열의 짝수 위치들에 있는 1의 개수가 홀수 위치들에 있는 1의 개수와 같을 때 **공정하다**고 부른다. 예를 들어 수열 011011은 공정하지만, 수열 011101은 공정하지 않다.

길이가 홀수인 0과 1의 수열이 주어진다. 이 수열에서 한 자리의 숫자를 하나 제거하는 것이 허용된다. 수열을 공정하게 만들려면 어느 자리를 제거해야 하는가?

예를 들어 수열 0111011에서는 두 번째 숫자를 지우면 된다.

### 입력 형식

입력 파일에는 한 줄이 있다. 이 줄에는 **홀수 길이**의 0과 1의 수열이 주어진다. 수열의 길이는 최대 200,001이다.

### 출력 형식

공정한 수열을 만들기 위해 제거해야 하는 수열의 자리 번호(1부터 시작)를 하나 출력하라.

만약 공정한 수열을 만들 수 없다면 0을 출력하고, 정답이 여러 개이면 그중 아무 것이나 출력하면 된다.

### 예제

표준 입력(stdin)	표준 출력(stdout)
0111011	2

## E. 측정기 놀이

은규는 동아리방 책상 위에서 특별한 제도 도구를 발견했는데, 일반적인 컴퍼스와 비슷하게 생긴 측정기였다. 이 측정기는 일반 컴퍼스와 달리 양쪽 다리에 모두 바늘이 달려 있다.

은규는 격자무늬 종이를 가져와서 측정기의 두 바늘 사이 거리를 일정하게 고정한 뒤 종이에 바늘을 꽂기 시작했다. 매번 측정기를 종이에 꽂을 때, 두 바늘 모두를 동시에 찍었고, 바늘 구멍은 항상 격자선이 교차하는 점에 생겼다. 또한, 한 번 만들어놓은 같은 구멍에 측정기를 여러 번 꽂을 수도 있었다.

저녁이 되어 민성이는 은규가 가지고 놀던 종이를 보게 되었고, 측정기의 두 바늘 사이 거리가 얼마였는지 알아내고자 했다. 민성이가 알고 있는 것은 측정기로 뚫린 모든 구멍의 좌표뿐이다.

민성이를 위해 은규가 설정했을 수 있는 바늘 사이의 거리를 구해보자.

### 입력 형식

첫 번째 줄에는 구멍의 개수  $n$ 이 주어진다. ( $2 \leq n \leq 1000$ )

다음  $n$ 개의 줄에는 구멍의 좌표  $x, y$ 가 각각 공백으로 구분되어 주어진다. ( $0 \leq |x|, |y| \leq 10^4$ )

### 출력 형식

첫 번째 줄에는 가능한 서로 다른 거리의 개수  $k$ 를 출력한다.

이어서  $k$ 개의 줄에 각각 가능한 거리를 출력한다. 거리는 **오름차순**으로 출력해야 하며, 정답과 절대 오차가  $10^{-9}$  이하일 시 정답으로 판정된다.

### 예제

표준 입력(stdin)	표준 출력(stdout)
4	2
0 0	1.0
1 1	1.4142135623730951
1 0	
0 1	

## F. AREA 998244353

SCCC 민방위 본부는 AREA 998244353의 핵 공격 대비 구조 계획을 새로 세우기로 했다. 이 지역의 모든  $n$ 개의 마을은 하나의 직선 도로를 따라 위치해 있다. 또한, 같은 도로를 따라  $m$ 개의 방공호도 위치해 있으며, 핵 공격 시 주민들은 이곳으로 대피할 수 있다.

핵 경보 시 구호가 최대한 효율적으로 이루어지기 위해서는, 각 마을마다 가장 가까운 방공호를 정해야 한다.

### 입력 형식

첫 번째 줄에는 마을의 개수  $n$ 이 주어진다. ( $1 \leq n \leq 10^5$ )

두 번째 줄에는  $n$ 개의 서로 다른 정수가 주어지며,  $i$ 번째 정수는 도로의 시작점으로부터  $i$ 번째 마을까지의 거리를 나타낸다.

세 번째 줄에는 방공호의 개수  $m$ 이 주어진다. ( $1 \leq m \leq 10^5$ )

네 번째 줄에는  $m$ 개의 서로 다른 정수가 주어지며,  $i$ 번째 정수는 도로의 시작점으로부터  $i$ 번째 방공호까지의 거리를 나타낸다.

모든 위치를 나타내는 숫자는 도로의 시작지점에서 떨어진 정도를 나타낸다.

모든 거리는 양수이며  $10^9$ 를 넘지 않는다. 마을과 방공호가 같은 지점에 위치할 수도 있다.

### 출력 형식

$n$ 개의 마을에 대해 가장 가까운 방공호의 인덱스를 각각 출력한다. 방공호는 입력에서 주어진 순서대로 1부터  $m$ 까지 번호가 매겨져 있다.

만약 가능한 답이 여러가지라면, 그 중 아무거나 하나를 출력한다.

### 예제

표준 입력(stdin)	표준 출력(stdout)
4 1 2 6 10 2 7 3	2 2 1 1



## G. 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.

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

## H. 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