



UCPC
greetings

22/23

Good Bye, BOJ 2022!

Official Problemset

주최 전국 대학생 프로그래밍 대회 동아리 연합

후원 STARTLINK



HYUNDAI
MOBIS

Presto

HYUNDAI
AutoEver

Proudly Supported By:

leejseo.com

한빛미디어
Hanbit Media, Inc.

문제 목록

문제지에 있는 문제가 총 8문제가 맞는지 확인하시기 바랍니다.

- A** 2022년이 아름다웠던 이유
- B** 나무 블록 게임
- C** 데이터 순서 복원
- D** 이미지 보정 작업
- E** 리그전
- F** 전설의 고대 광산 탈출
- G** 크루스칼 알고리즘
- H** Maxtrix

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 2022년이 아름다웠던 이유

시간 제한 1 초
메모리 제한 1024 MB

Good Bye, BOJ 2022! 대회는 2022년의 끝을 기념하는 알고리즘 문제해결 대회이다. leejseo라는 핸들을 사용하는 종서는 자신의 22번째 생일인 2022년 12월 31일에 Good Bye, BOJ 2022!를 개최하기로 결심했다.

정휘는 Good Bye, BOJ 2022!가 개최된다는 소식에 기뻐 제목을 뚫어져라 보다가, 2022가 정말 아름다운 수라는 사실을 깨달았다.

자기 자신을 제외한 약수들의 합이 자기 자신보다 작은 수를 부족수, 자기 자신보다 큰 수를 과잉수, 자기 자신과 같은 수를 완전수라고 하자.

그러면 2022는 과잉수이면서 2022의 자기 자신을 제외한 모든 약수는 모두 부족수이거나 완전수이다. 과잉과 부족의 조화라니, 이 얼마나 아름다운 수인가! 다음에 이런 년도가 오려면 2044년이 되어야 한다.

양의 정수 n 이 주어진다. n 이 과잉수이면서 n 을 제외한 n 의 모든 약수가 부족수이거나 완전수인지 판별하는 프로그램을 작성하여라.

입력

첫째 줄에 테스트 케이스의 개수 T 가 주어진다.

이후 T 개의 줄에 걸쳐 테스트 케이스가 한 줄에 하나씩 주어진다. 각 테스트 케이스는 한 줄로 구성되며, 각각 한 개의 양의 정수 n 이 주어진다.

출력

각 테스트 케이스에 대해, n 이 과잉수이면서 n 을 제외한 n 의 모든 약수가 부족수이거나 완전수라면 **Good Bye**, 그렇지 않다면 **BOJ 2022**를 한 줄에 하나씩 차례로 출력하여라.

제한

- $1 \leq T \leq 500$
- $1 \leq n \leq 5000$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 2022 2023 2044	Good Bye BOJ 2022 Good Bye



Good Bye, BOJ 2022!

이 페이지는 공백입니다

문제 B. 나무 블록 게임

시간 제한 1 초
메모리 제한 1024 MB

올해도 집에서 쓸쓸히 혼자 새해를 맞이하는 주원이는 제야의 종이 울리기 전까지 심심함을 달랠 무언가를 찾고 있다.

주원이는 근처에 굴러다니는 N 개의 나무 블록을 발견했다. 각 나무 블록에는 음이 아닌 정수가 한 개씩 적혀있었다. 심심한 주원이는 나무 블록들을 가지고 혼자서 할 수 있는 간단한 게임을 떠올렸다.

처음에 N 개의 나무 블록은 N 개의 주머니에 한 개씩 들어있다. 이때 주원이는 다음 작업을 원하는 만큼 반복할 수 있다.

- 서로 다른 두 주머니를 고른 다음 둘을 하나로 합친다

예를 들어, 첫 번째 주머니에 2가 적힌 나무 블록과 5가 적힌 나무 블록이, 두 번째 주머니에 3이 적힌 나무 블록이 들어있다고 하자. 이 두 주머니를 합치면 각각 2, 3, 5가 적힌 나무 블록 총 세 개가 하나의 주머니에 들어 있게 된다. 따라서 주머니의 총 개수는 하나 줄어든다.

게임의 모든 과정이 끝난 뒤 남아있는 주머니의 개수가 K 개라고 할 때, 각각에 대해 들어있는 나무 블록에 적힌 수의 평균을 구하자. 각 주머니에서 계산한 K 개의 평균들을 오름차순으로 나열했을 때 $\left\lfloor \frac{K+1}{2} \right\rfloor$ 번째에 위치한 값이 주원이의 점수가 된다.

제야의 종이 울리기 전에 주원이가 게임을 마스터할 수 있도록 나무 블록의 정보가 주어지면 얻을 수 있는 점수의 최댓값을 구해주는 프로그램을 만들어주자.

입력

첫째 줄에 나무 블록의 개수 N 이 주어진다.

둘째 줄에 각 나무 블록에 적혀있는 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

출력

게임에서 얻을 수 있는 가장 큰 점수를 출력한다. 절대/상대 오차는 10^{-6} 까지 허용한다.

제한

- $1 \leq N \leq 100000$
- $0 \leq A_i \leq 10000$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 0 2 2	2
2 1 2	1.500000

설명

첫 번째 입출력 예시에서 각 주머니에 들어 있는 나무 블록에 적힌 수들을 중괄호 $\{\}$ 에 넣어 표시하자.

아무런 작업을 하지 않는다면 주머니는 $\{2\}, \{0\}, \{2\}, \{2\}$ 와 같이 되고, 평균은 각각 2, 0, 2, 2가 되기 때문에 2점을 얻는다.

첫 번째와 두 번째 주머니를 합친다면 $\{2, 0\}, \{2\}, \{2\}$ 가 되어 평균은 각각 1, 2, 2가 되기 때문에 2점을 얻는다.

이후 첫 번째와 두 번째를 합친 주머니와 세 번째 주머니를 합친다면 $\{2, 0, 2\}, \{2\}$ 가 되어, 평균은 각각 $\frac{4}{3}, 2$ 가 되기 때문에 $\frac{4}{3}$ 점을 얻는다.

문제 C. 데이터 순서 복원

시간 제한 1 초
메모리 제한 1024 MB

Presto Labs는 High-Frequency trading을 하는 기업으로, Low-Latency In-House Trading Platform을 개발하는 등 트레이딩 과정의 각 단계에서 발생하는 지연 시간을 추적하고, 최소화하기 위해 많은 노력을 하고 있다. Presto Labs는 찬솔이에게 지연 시간과 관련한 분석 업무를 맡기면서 단계별로 발생하는 지연 시간의 대소 관계와 관련된 데이터를 함께 제공해주었다.

원본 데이터는 1단계부터 N 단계까지 각 단계의 지연 시간이 적은 순으로 배열되어 있다. 그러나, 찬솔이의 장난기 많고 짓궂은 친구 정휘가 원본 데이터를 찬솔이의 컴퓨터에서 삭제해버렸다. 대신 정휘는 자신이 갖고 있는 원본 데이터를 바탕으로 찬솔이에게 변형된 데이터 세 개를 주면서

“어차피 많이 변형시키지는 않았으니, 너무 걱정하지는 마. 너라면 충분히 원본 데이터를 복원해낼 수 있을 거야.”라고 말했다. 그러면서, 정휘는 자신이 변형한 모든 데이터는 다음과 같은 규칙을 만족하도록 훼손시켰다는 사실 또한 알려줬다:

- 각 변형된 데이터는 정확히 하나의 단계가 원래 위치보다 앞으로 가도록 변형되었다.
- 각 변형된 데이터마다 앞으로 옮겨진 단계는 다르다.

찬솔이를 도와 변형된 데이터로부터 원본 데이터(즉, 단계별 지연 시간의 순서)를 복원해 보자.

입력

첫째 줄에 단계의 수 N 이 주어진다.

둘째 줄에 변형된 데이터 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

셋째 줄에 변형된 데이터 B_1, B_2, \dots, B_N 이 공백으로 구분되어 주어진다.

넷째 줄에 변형된 데이터 C_1, C_2, \dots, C_N 이 공백으로 구분되어 주어진다.

최소한 하나 이상의 방법이 존재하도록 입력이 주어진다.

출력

원본 데이터를 출력한다. 가능한 방법이 여러 개라면 그 중 아무거나 출력한다.

제한

- $4 \leq N \leq 3000$
- $1 \leq A_i, B_i, C_i \leq N$
- 모든 A_i 들은 서로 다르다.
- 모든 B_i 들은 서로 다르다.
- 모든 C_i 들은 서로 다르다.

입출력 예시

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

설명

첫 번째 입출력 예시에서 가능한 답 중 하나는 2, 3, 5, 6, 1, 4 이다. 이 경우에 데이터는 정렬에 의해 다음과 같이 변형되었다.

첫 번째 데이터는 단계 3이 단계 2보다 앞으로 가도록 변형되어 3, 2, 5, 6, 1, 4가 되었다.

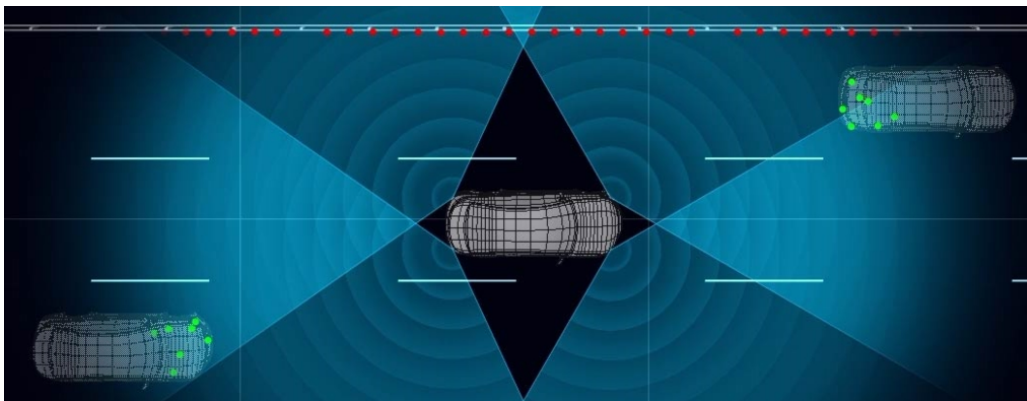
두 번째 데이터는 단계 4가 단계 6보다 앞으로 가도록 변형되어 2, 3, 5, 4, 6, 1가 되었다.

세 번째 데이터는 단계 6이 단계 3보다 앞으로 가도록 변형되어 2, 6, 3, 5, 1, 4가 되었다.

문제 D. 데이터 순서 복원

시간 제한 3 초
메모리 제한 1024 MB

현대모비스는 글로벌 자동차 부품 기업으로 자율주행, 커넥티비티, 전동화 분야 등 스마트 모빌리티 솔루션 연구에 강점을 갖고 있다. 자율주행 및 주행보조 성능 개선 등 소프트웨어 분야에 대한 활발한 기술 개발을 진행하고 있는 현대모비스는 차량의 카메라로 얻은 이미지를 처리하는 소프트웨어를 개발하고 있다. 특히 야간, 악천후 등의 상황에서 이미지가 선명하지 않을 수 있는데, 소프트웨어 기반의 후처리를 통해 이미지를 보정하는 것은 중요한 일이다.



차량의 카메라가 얻은 이미지 정보는 세로로 N 칸, 가로로 M 칸으로 분할된 총 NM 개의 구역으로 나누어 표현할 수 있다. 각 구역 내의 이미지 정보가 얼마나 선명한지를 정수로 나타내고 이를 **선명도**라 부르자.

한 구역의 이미지를 보정하면 해당 구역의 선명도를 X 로 만들 수 있다. 물론 모든 구역을 보정하면 좋겠으나, 운전자에게 이미지가 보여지기까지 시간이 많이 걸리면 불편하므로 K 개 이하의 구역만 보정하고자 한다.

인접한 두 구역의 선명도 차이가 크다면 운전자가 보기에 이미지가 어색하기 때문에, 인접한 두 구역의 선명도의 가장 큰 차이가 최소가 되어야 한다. 만약 두 구역이 서로 변을 공유한다면, 두 구역이 인접하다고 한다.

현대모비스를 위해 K 개 이하의 구역을 보정하여 만들 수 있는 인접한 두 구역의 선명도의 가장 큰 차이의 최솟값을 구해보자.

입력

첫째 줄에 이미지 정보의 세로 칸 개수 N , 가로 칸 개수 M , 보정할 수 있는 구역의 개수 K , 보정했을 때의 선명도 X 가 공백으로 구분되어 주어진다.

둘째 줄부터 N 개의 줄에는 각각 M 개의 정수 $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ 이 공백으로 구분되어 주어진다. $a_{i,j}$ 는 이미지 정보의 위에서 i 번째, 왼쪽에서 j 번째 구역의 선명도를 의미한다.

출력

이미지를 보정해서 만들 수 있는 인접한 두 구역의 선명도의 가장 큰 차이의 최솟값을 출력하라.

제한

- $1 \leq N, M \leq 500$
- $0 \leq K \leq NM$
- $1 \leq X \leq 10^{18}$

- $0 \leq a_{i,j} \leq X$

입출력 예시

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

문제 E. 리그전

시간 제한 2 초
메모리 제한 1024 MB

리그전에서는 1번 팀부터 n 번 팀까지 총 n 개의 팀이 다른 모든 팀과 한 번씩 맞붙는다. 경기 결과는 승리, 패배, 혹은 무승부 중 하나이고 이기면 승점 a 점, 비기면 b 점, 지면 c 점을 획득한다. 단, 일반적인 리그전의 승점 계산 방식과는 다르게 이 문제에서 a, b, c 의 대소 관계는 정해져 있지 않다. 예를 들어, 이겼을 때 얻는 점수가 졌을 때 얻는 점수보다 적을 수 있다.

팀의 순위는 승점이 높은 순으로 정하되, 만약 승점이 같으면 팀 번호가 작은 팀이 더 높은 순위이다. 모든 경기가 끝난 뒤 k 등 팀의 승점으로 가능한 값 중 최댓값과 최솟값을 구하는 프로그램을 작성하여라.

입력

첫 번째 줄에 테스트 케이스의 개수 T 가 주어진다.

다음 T 개의 줄에 각 줄마다 a, b, c, n, k 가 공백을 사이에 두고 주어진다.

출력

T 개의 줄에 각 케이스마다 k 등 팀의 승점의 최댓값과 최솟값을 순서대로 출력한다.

제한

- $1 \leq T \leq 100000$
- $0 \leq a, b, c \leq 10^9$
- $1 \leq k \leq n \leq 10^9$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5	9 3
3 1 0 4 1	7 2
3 1 0 4 2	42 42
7 7 7 7 3	37 24
4 7 8 6 4	17 0
2 0 5 5 2	



WCPC
greetings

22/23

Good Bye, BOJ 2022!

설명

	1	2	3	4	승점
1		승	승	승	9
2	패		승	승	6
3	패	패		승	3
4	패	패	패		0

	1	2	3	4	승점
1		무	무	무	3
2	무		무	무	3
3	무	무		무	3
4	무	무	무		3

	1	2	3	4	승점
1		무	승	승	7
2	무		승	승	7
3	패	패		승	3
4	패	패	패		0

	1	2	3	4	승점
1		승	승	승	9
2	패		무	무	2
3	패	무		무	2
4	패	무	무		2

문제 F. 전설의 고대 광산 탈출

시간 제한 3 초
메모리 제한 1024 MB

당신은 전설 속의 고대 광산을 탐험하고 있다. 이 광산 곳곳에는 값비싼 희귀 광석이 넘쳐나지만, 광석을 수집하러 들어갔다가 살아 나온 사람은 단 한 명도 없었다고 전해진다.

직접 방문한 전설의 고대 광산에는 하나의 직선 레일이 놓여 있었다. 당신은 이 레일을 따라 이동하기로 했다. 전설에서 들은 대로 광산 안에는 희귀 광석들이 넘쳐났고, 레일 옆 곳곳에는 고대의 광부들이 광석에 담아 둔 수레가 놓여 있었다. 당신은 총 N 개의 수레에 대한 정보를 수첩에 기록해 두었다. 당신이 기록한 정보는 구체적으로 다음과 같다.

- i 번째 수레는 입구로부터 거리가 x_i 인 곳에 위치한다.
- i 번째 수레에는 광석 주머니가 하나 담겨 있다. 이 주머니의 질량은 m_i 이고, 주머니에 담긴 광석의 가치는 c_i 이다.

당신이 광산 입구로부터 거리가 X 인 위치에 도달했을 때, 광산 안에서 엄청난 굉음이 들리기 시작했다. 당신은 T 만큼의 시간이 지나면 광산 전체가 무너져 내릴 것이라는 사실을 직감했고, 근처에 있던 빈 수레를 레일에 얹은 다음 그 위에 올라탔다. 당신과 빈 수레의 질량의 합은 M 이다. 당신은 수레를 타고 V 의 속도로 입구를 향해 달리기 시작했다.

레일은 평평한 지형 위에 놓여 있으며 마찰이 없기 때문에, 당신이 달리는 동안 당신이 탄 수레의 운동량(질량과 속도의 곱)은 MV 로 일정하게 유지될 것이다. 이 속도라면 탈출에 성공할 수 있을 거라는 확신에 찬 당신은, 조금 더 욕심을 내서 달리는 도중 만나는 수레에 담긴 광석 주머니 몇 개를 담아 가기로 결심했다. 당신은 손이 빠르기 때문에 순식간에 옆 수레에서 광석 주머니를 가져올 수 있다. 하지만 주머니를 담을 때마다 당신이 탄 수레는 무거워지고, 운동량 보존의 법칙에 의해 당신이 탄 수레는 느려지게 된다. 구체적으로, 당신이 주머니를 담은 이후의 질량을 M' , 그 때의 속도를 V' 라고 하면, $M'V' = MV$ 가 성립한다.

당신의 최우선 목표는 전설의 고대 광산을 살아 나오는 것이다. 시간 T 이내에 광산 입구에 도달해야 하며, 정확히 T 만큼의 시간이 걸려도 괜찮다. 또한 당신은 수레에 담은 광석의 총가치도 최대화하고자 한다. 당신이 전설의 고대 광산을 살아 나오는 것이 가능하다면, 가지고 나올 수 있는 광석의 총가치는 최대 얼마일지 계산하시오.

입력

첫 번째 줄에 X, T, M, V, N 이 공백으로 구분되어 주어진다.

두 번째 줄부터 N 개의 줄만큼 x_i, m_i, c_i 가 공백으로 구분되어 주어진다.

출력

당신이 전설의 고대 광산을 탈출하는 것이 가능하다면, 당신이 가지고 나올 수 있는 광석의 총가치가 최대 얼마인지 출력한다.

당신이 전설의 고대 광산을 탈출하는 것이 불가능하다면, -1 을 출력한다.

제한

- $1 \leq X, T, M, V, N \leq 100$
- $1 \leq x_i \leq X$
- $1 \leq m_i, c_i \leq 10000$

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 5 5 5 3 5 10 3 5 10 7 5 10 2	10
5 2 5 2 3 5 10 3 5 10 7 5 10 2	-1

문제 G. 크루스칼 알고리즘

시간 제한 2 초
메모리 제한 1024 MB

정휘는 알고리즘 강의에서 그래프의 최소 신장 트리를 구하는 알고리즘인 크루스칼 알고리즘에 대해 배웠다. 크루스칼 알고리즘은 다음과 같이 동작한다.

1. 간선을 가중치 오름차순으로 정렬한다. 이때 두 간선의 가중치가 동일하다면 두 간선 중 어떠한 것이 앞에 와도 상관없다.
2. 최소 신장 트리를 이루는 간선의 집합 F 를 공집합으로 초기화한다. ($F := \emptyset$)
3. 간선을 차례대로 보면서, 만약 간선 e 의 양 끝점이 F 의 간선들을 통해 연결되어 있지 않다면 e 를 F 에 추가한다. 즉, $F \cup \{e\}$ 에 사이클이 없다면 F 에 e 를 추가한다.

연결 무향 가중치 그래프가 주어졌을 때 알고리즘이 최소 신장 트리를 구한다는 것은 어렵지 않게 증명할 수 있다.

정휘는 (1)에서 가중치가 같은 간선을 정렬하는 기준이 없다는 점이 마음에 들지 않는다. 같은 그래프가 주어졌도 다른 결과가 나올 수 있다는 것은 용납할 수 없는 일이다. 정휘는 자신의 분노를 수치화하기 위해 얼마나 다양한 결과가 나올 수 있을지 계산하려고 했지만, 알고리즘 강의를 끝날 때까지 그 방법을 찾지 못해서 여러분에게 문제로 내기로 했다.

N 개의 정점과 M 개의 간선으로 구성된 단순 연결 무향 가중치 그래프가 주어진다. 크루스칼 알고리즘을 이용해 최소 신장 트리를 구할 때, F 에 간선을 추가하는 방법의 수를 구하는 프로그램을 작성해 보자. F 에 추가된 간선들의 집합이 다르거나, 간선들을 추가하는 순서가 다르다면 다른 방법으로 센다.

입력

첫째 줄에 정점의 개수 N 과 간선의 개수 M 이 공백으로 구분되어 주어진다.

둘째 줄부터 M 개의 줄에 걸쳐 두 간선이 연결하는 정점 번호 u_i, v_i 와 간선의 가중치 w_i 가 주어진다.

출력

F 에 간선을 추가하는 순서로 가능한 경우의 수를 $998244353 (= 119 \times 2^{23} + 1)$ 으로 나눈 나머지를 출력한다.

제한

- $2 \leq N \leq 400$
- $N - 1 \leq M \leq N(N - 1)/2$
- $1 \leq u_i < v_i \leq N$
- $1 \leq w_i \leq 10^9$
- $i \neq j$ 이면 $(u_i, v_i) \neq (u_j, v_j)$
- 모든 정점은 간선을 통해 서로 직/간접적으로 연결되어 있다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 4 1 2 1 1 3 1 1 4 1 1 5 1	24
5 4 1 2 1 1 3 2 1 4 3 1 5 4	1
3 3 1 2 1 1 3 1 2 3 1	6

문제 H. Maxtrix

시간 제한 2 초
메모리 제한 1024 MB

길이가 N 인 두 정수 배열 A, B 가 주어진다. 이때 $N \times N$ 크기의 행렬 C 가 다음과 같이 정의된다.

$$C_{ij} = A_i + B_j - ij$$

1 이상 N 이하의 모든 정수 k 에 대해 각각 $\max_{1 \leq i \leq k \leq j \leq N} C_{ij}$ 를 계산하라.

입력

첫 번째 줄에 두 배열의 길이를 나타내는 정수 N 이 주어진다.

두 번째 줄에는 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

세 번째 줄에는 B_1, B_2, \dots, B_N 이 공백으로 구분되어 주어진다.

출력

N 개의 줄에 걸쳐 답을 출력한다.

제한

- $1 \leq N \leq 250000$
- A, B 의 모든 원소의 절댓값은 10^{18} 이하.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 3 2 -2 6 1 3 -3 0	4 4 -1 -1
1 1000000000000000000 -1000000000000000000	-1



Good Bye, BOJ 2022!

이 페이지는 공백입니다