

대회 안내

1. 대회 관련

- 각 문제를 해결할 때마다 ({대회 후 경과시간} + {해당 문제 오답 횟수x20분})의 패널티가 가산됩니다.
- 순위는 문제를 많이 푼 순 > 패널티가 적은 순 > 마지막 정답이 빠른 순으로 정해집니다.
- 문의하기 기능을 사용해 문제와 직접적으로 관련된 사항에 대해 질문할 수 있습니다.

2. 제출관련

- 제출시 문제 번호와 사용 언어를 정확히 선택해주세요. 참가자의 실수에 대해 책임지지 않습니다.
- 제출한 코드는 채점 현황에서 확인할 수 있습니다.

3. 채점 관련

- 표준 입출력 이외의 파일을 읽거나 쓰려고 시도하는 경우 런타임 에러로 처리하여 오답으로 인정합니다.
- 병렬연산이나 스레드를 이용하면 안됩니다. 모든 프로그램은 단일 스레드 프로그램이어야 합니다.
- 시스템에 영향을 주는 기능(exit, sleep 등)을 사용하면 무조건 오답처리 합니다.
- 채점 입력 데이터 모두에 대해 올바른 답을 출력하여야 합니다. 부분 점수는 없습니다.
- 예제 데이터는 하나의 예제 일 뿐입니다. 실제 채점은 수 많은 비공개 데이터들로 이루어집니다.
- 각 채점 결과의 의미는 아래와 같습니다. 감독관은 오답의 이유를 절대 알려드리지 않습니다.
 - **기다리는 중** : 채점을 기다리는 중입니다. 조금 기다리면 채점 결과를 받을 수 있습니다.
 - **재채점을 기다리는 중** : 재채점을 기다리는 중입니다.
 - **채점 준비중** : 채점 프로그램이 채점을 하기 위해 여러가지 작업을 수행하는 중 입니다. 소스 코드 컴파일, 실행 환경 세팅 등이 포함됩니다.
 - **채점중** : 채점을 하는 중입니다.
 - **맞았습니다!!** : 제출한 프로그램이 모든 데이터를 맞은 경우에 받게 됩니다.
 - **출력 형식이 잘못되었습니다** : 답을 올바르게 구했으나, 문제에 나와있는 출력 형식과 다른 경우입니다. 다음은 이 결과를 받게 되는 일부 예시입니다.
 - 줄 뒤에 의미 없는 공백을 출력하는 경우. 단, 공백 한 칸은 무시하고 채점합니다. 예를 들어, "1 2"를 출력해야 하는 경우에 "1 2"와 "1 2 "는 맞았습니다!!를 받게 됩니다. "1 2 "는 공백 두 칸을 뒤에 출력해서, " 1 2"는 첫 문자가 공백이라서, "1 2"는 두 수를 공백 두 칸으로 구분했기 때문에, **출력 형식이 잘못되었습니다.**를 받게 됩니다. 여기서 "는 공백을 보기 위해 편의상 삽입한 기호입니다.
 - 의미없는 빈 줄을 삽입하는 경우. 예를 들어, 1부터 N까지 한 줄에 하나씩 출력해야 하는 경우에 1부터 N을 한줄에 하나씩 공백으로 구분해서 출력하는 경우는 **맞았습니다!!**를 받게 됩니다. 이것은 사이트가 생겼을 때 부터 있던 채점 시스템의 버그로 수정될 예정입니다. 각 줄의 사이에 빈 줄을 출력하는 경우에는 **출력 형식이 잘못되었습니다.**를 받게 됩니다.
 - **틀렸습니다** : 출력 결과가 정답과 다른 경우입니다.
 - **시간 초과** : 프로그램이 제한된 시간 이내에 끝나지 않은 경우입니다. 시간 제한을 초과하면 실행을 즉시 중단합니다. 따라서, 정답이 맞았는지 틀렸는지는 알 수 없습니다.
 - **메모리 초과** : 프로그램이 허용된 메모리보다 많은 메모리를 사용한 경우입니다. 와 마찬가지로, 메모리 제한을 초과하면 실행을 즉시 중단합니다.
 - **출력 초과** : 프로그램이 너무 많은 출력을 하는 경우에 발생합니다. 이 결과는 와 같은 의미를 갖습니다. 정확하게는 미리 구해놓은 정답 파일 크기의 2배를 넘어가면 이 결과를 받게 됩니다. 스

페셜 저지는 미리 구해놓은 정답 파일이 없을 수도 있기 때문에, 이 결과를 받을 수 없습니다. 만약, 출력하는 부분이 무한 루프에 빠진 경우, 시간을 초과하기 전에 결과를 받을 수 있습니다.

- **런타임 에러** : 프로그램이 비정상적으로 종료한 경우입니다. Exit code가 0이 아닌 경우, **segmentation fault**를 받은 경우가 대표적입니다.
- **컴파일 에러** : 컴파일에 실패한 경우입니다. 경고(Warning)가 있어도 컴파일에 성공하면 이 결과를 받지 않습니다. 결과를 클릭하면 컴파일 에러 메시지를 볼 수 있습니다.

4. C/C++ 관련

- 모든 입출력은 stdin, stdout을 통한 표준 입출력만 사용됩니다.
- GCC 컴파일러를 통해 채점됩니다. 비표준 MS C/C++ 라이브러리를 사용할 경우 컴파일 에러가 발생할 수 있습니다.
- C++의 cin/cout은 scanf/printf에 비해 속도가 현저히 느립니다. 채점 시간에 반영되므로 후자의 이용을 권장합니다.

5. Java 관련

- 소스코드에 Package 선언문을 절대로 넣지 마세요. 오답처리 됩니다.
- 모든 입출력은 system.in, System.out을 통한 표준 입출력만 사용됩니다.
- Java의 특성상 문자열 처리 시 StringBuilder를 활용하는 편이 속도면에서 매우 유리합니다.
- system.in 입력 시 Scanner보다 BufferedReader와 InputStreamReader를 활용하는 편이 훨씬 속도가 빠릅니다.

6. Python 관련

- `arr = list(map(int, input().split()))`
- 위와 같이 사용하면 한 줄 단위로 입력을 받게 되고, 각 줄에서 띄어쓰기로 구분된 값들이 개수에 상관 없이 전부 arr에 들어가게 됩니다. 이후 기존 list를 다루듯이 사용하시면 됩니다.

C/C++ 레퍼런스 사이트 : <http://en.cppreference.com/w/>

Java 레퍼런스 사이트 : <http://docs.oracle.com/javase/7/docs/>

Python 레퍼런스 사이트 : <http://python.org/>

Problem A

나는 누구인가

시간 제한 : 1 Second

당신은 지금 코딩을 하고있다. 그럼 당신은 어떤 사람인가.

입력

없다.

출력

“I’m Sexy”를 출력한다.

예제 입력

예제 출력

	I’m Sexy
--	----------

Problem B

명장 남정훈

시간 제한 : 1 Second

오늘도 어김없이 피시방에서 피파를 하고 있는 정훈이는 큰 고민에 빠졌다. 자신에 팀에 있는 선수들의 주 사용 발(Main Foot)이 적절하게 나눠져있지 않기 때문이다. 정훈이의 팀에는 L명의 왼발잡이 선수와, R명의 오른발잡이 선수, A명의 양발잡이 선수가 존재한다. 양발잡이 선수는 오른발잡이 선수처럼 생각해도 되고, 왼발잡이 선수로도 생각해도 된다. 11명의 주전선수를 뽑기 전에 정훈이가 가진 팀원을 정리하려고 한다. 정훈이는 왼발잡이와 오른발잡이 선수의 수를 같게 만든 다음, 나머지 인원은 방출하려 한다. 정훈이가 정리하고 난 다음, 팀에 잔류하고 있는 선수의 최대 수를 구해보자.

입력

첫 번째 줄에 왼발잡이 선수의 수 L, 오른발잡이 선수의 수 R, 양발잡이 선수의 수 A가 주어진다. 각 숫자는 0이상 100이하이다.

출력

첫 번째 줄에 최대 잔류 인원 수를 출력한다.

예제 입력

1 5 2	6
-------	---

예제 출력

예제 입력

7 7 7	20
-------	----

예제 출력

Problem C

매직스퀘어

시간 제한 : 1 Second

수학에서 행렬(行列, matrix)은 수나 기호, 수식 등을 네모꼴로 배열한 것으로, 가로줄을 행(row), 세로줄을 열(column)이라고 한다. 특히 행과 열의 개수가 같은 행렬을 정방행렬(square matrix)이라고 한다.

매직스퀘어(magic square, 마방진, 魔方陣)는 다음 2 가지 성질을 만족하는 정방행렬이다.

1. 행렬의 원소는 1부터 $n \times n$ (n의 제곱) 범위의 정수로 구성하며, 중복되는 수가 없이 모두 달라야 한다.
2. 행의 합, 열의 합, 2개의 대각선의 수열의 합은 모두 같으며, 합은 다음 식으로 구한다.

$$n * (n^2 + 1) / 2$$

즉, $n=2$ 이면 합은 5 이고, $n=3$ 이면 합은 15 이고, $n=4$ 이면, 합은 34 이다.

예를 들어, 다음은 매직스퀘어의 예이다. ($n=2$ 인 매직스퀘어는 없다.)

		4 14 15 1	11 24 7 20 3
2 7 6		9 7 6 12	4 12 25 8 16
9 5 1	5 11 10 8	17 5 13 21 9	
4 3 8	16 2 3 13	10 18 1 14 22	
		23 6 19 2 15	

정수 리스트의 리스트(또는 정수 배열의 배열)로 표현한 임의의 정방행렬을 받아서, 매직스퀘어인지 확인하는 프로그램을 작성하시오.

입력

첫째 줄에 $N(3 \leq N < 100)$ 가 주어지고 둘째 줄부터 N개의 줄에 걸쳐 N개의 숫자가 주어진다. N개의 수의 범위는 10이상 $n * (n^2 + 1) / 2$ 이다.

출력

입력된 N개의 숫자가 나타내는 정방행렬이 매직스퀘어라면 TRUE, 아니라면 FALSE를 출력한다.

예제 입력

예제 출력

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

Problem D

삼각

시간 제한 : 1 Second

N층의 정삼각형이 주어졌을때, 크고 작은 정삼각형의 개수가 몇개인지 알아보자.
N이 2일 경우 정답은 5이다.



입력

첫 번째 줄에 정삼각형 탑의 층 N이 주어진다. ($1 \leq N \leq 10,000$)

출력

첫 번째 줄에 크고 작은 삼각형의 개수의 합을 출력한다.

예제 입력

2	5
---	---

예제 출력

예제 입력

10	315
----	-----

예제 출력

Problem E

청기 백기

시간 제한 : 1 Second

소프트웨어융합대학 학생회에서 주최한 소용체전에서 청기 백기 뒤집기 게임이 한창이다. 소프트웨어학부, ICT 융합학부가 번갈아가면서 게임을 진행하는 중이다. 게임의 규칙은 간단하다. 게임을 진행할 차례인 학부에서 출전한 선수들 N 명이 존재한다. 학생들의 앞 탁자에는 N 개의 깃발이 청색이 위로 백색이 아래로 보이도록 놓여있다. 이 때 출전한 선수 중 첫 번째 선수는 N 개의 깃발 중 1의 배수에 해당하는 번호의 깃발을 뒤집어 놓는다. 다음 두 번째 선수는 N 개의 깃발 중 2의 배수에 해당하는 번호의 깃발을 뒤집어 놓는다. i 번째 선수는 i 의 배수에 해당하는 번호의 깃발을 뒤집고, N 번째 선수까지 진행하면 끝이 난다. 그렇다면 이 게임에서 N 명의 선수가 참가하고 N 개의 깃발이 존재할 때, N 번째 선수까지 진행하여 완료된 상태에서 백색이 위로 놓여있는 깃발의 수가 몇 개인지 알아보자.

입력

첫 번째 줄에 출전한 학생의 수이자, 깃발의 개수인 $N(1 \leq N \leq 2,100,000,000)$ 이 주어진다.

출력

첫 번째 줄에 N 번째 선수까지 진행한 후의 상태의 깃발 중 백색이 위로 놓여있는 깃발의 수를 출력한다.

예제 입력 1

3	1
---	---

예제 출력 1

예제 입력 2

24	4
----	---

예제 출력 2

힌트

예제 입력 1의 경우 1, 2, 3번 깃발이 존재하고, 3명의 선수가 참가한다. 첫 번째 선수는 1의 배수의 번호를 가진 깃발을 뒤집는다. 초기에 청색이었던 깃발은 첫 번째 선수에 의해 모두 백기로 된다. 두 번째 선수는 2의 배수의 번호를 가진 깃발, 2번 깃발을 뒤집는다. 3개의 깃발은 백 청 백의 순서로 놓여있게 된다. 마지막 선수는 3의 배수의 번호를 가진 깃발, 3번 깃발을 뒤집는다. 마지막 깃발의 상태는 백 청 청이 되고, 따라서 백기의 수는 1이다.