



2020      The 0x10th

# Sogang Programming Contest

2020 서강대학교 프로그래밍 대회



## Master Official Problemset



## 대회 규칙

- 대회 중 제출한 소스코드는 채점 서버에 의해 자동으로 채점되며, 실시간으로 결과를 알 수 있습니다. 문제를 제출하였을 때 ‘맞았습니다!!’를 받으면 문제를 푼 것으로, 이외의 결과를 받으면 틀린 것으로 생각합니다.
- 문제를 풀 때마다 패널티 점수가 누적됩니다. 패널티 점수는 모든 맞은 문제에 대해, 대회 시작 시간부터 그 문제를 풀기까지 걸린 시간을  $t$ 분, 처음으로 문제를 맞기 직전까지 제출한 횟수를  $w$ 번이라고 할 때  $(t + 20w)$  점입니다.
- 순위는 푼 문제가 많은 순서대로, 푼 문제가 같을 경우에는 패널티 점수의 합이 적은 순서대로 결정됩니다.
- 사용 가능 언어는 C, C++, Java, Python 3, Kotlin입니다. 모든 문제는 출제진이 C++과 Java(혹은 Kotlin), PyPy 3으로 정답을 작성했음이 보장됩니다. 각 언어의 컴파일 옵션과 컴파일러 버전은 아래와 같습니다.

### C11 gcc (GCC) 10.2.0

컴파일 `gcc Main.c -o Main -O2 -Wall -lm -static -std=gnu11 -DONLINE_JUDGE -DBOJ`  
실행 `./Main`

### C++11 gcc (GCC) 10.2.0

컴파일 `g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++11 -DONLINE_JUDGE -DBOJ`  
실행 `./Main`

### C++14 gcc (GCC) 10.2.0

컴파일 `g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++14 -DONLINE_JUDGE -DBOJ`  
실행 `./Main`

### C++17 gcc (GCC) 10.2.0

컴파일 `g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++17 -DONLINE_JUDGE -DBOJ`  
실행 `./Main`

### C++20 gcc (GCC) 10.2.0

컴파일 `g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++20 -DONLINE_JUDGE -DBOJ`  
실행 `./Main`

### Java 8 openjdk 15 2020-09-15

컴파일 `javac -release 8 -J-Xms1024m -J-Xmx1920m -J-Xss512m -encoding UTF-8 Main.java`  
실행 `java -Xms1024m -Xmx1920m -Xss512m -Dfile.encoding=UTF-8 -XX:+UseSerialGC -DONLINE_JUDGE=1 -DBOJ=1 Main`

### Python 3 Python 3.8.2

컴파일 `python3 -c "import py_compile; py_compile.compile(r'Main.py')"`  
실행 `python3 Main.py`

### PyPy3 PyPy 7.3.2-alpha0 with GCC 7.3.1 20180303 (Red Hat 7.3.1-5) (Python 3.7.4)

컴파일 `python3 -c "import py_compile; py_compile.compile(r'Main.py')"`  
실행 `pypy3 Main.py`

### Kotlin/JVM kotlinc-jvm 1.4.10 (JRE 1.8.0\_201-b09)

컴파일 `kotlinc-jvm -J-Xms1024m -J-Xmx1920m -J-Xss512m -include-runtime -d Main.jar Main.kt`

실행 `java -Xms1024m -Xmx1920m -Xss512m -Dfile.encoding=UTF-8 -XX:+UseSerialGC -DONLINE_JUDGE=1 -DBOJ=1 -jar Main.jar`

- 시간 제한에 ‘추가 시간 있음’이 표시된 일부 문제들은 언어별로 추가 시간 및 메모리를 제공합니다. 변경하여 적용되는 시간 및 메모리 제한은 언어별로 다음과 같습니다.

- **Java 8, Kotlin/JVM:** 시간  $\times 2 + 1$ 초, 메모리  $\times 2 + 16$ MB

- **Python 3:** 시간  $\times 3 + 2$ 초, 메모리  $\times 2 + 32$ MB

- **PyPy3:** 시간  $\times 3 + 2$ 초, 메모리  $\times 2 + 128$ MB

- 네트워크 사용은 금지됩니다. 단, 솔루션을 제출하거나 언어 레퍼런스를 확인하는 것은 가능합니다. 접속이 허용된 사이트의 예는 다음과 같습니다.

**C/C++** <https://en.cppreference.com/w/>

**Java** <https://docs.oracle.com/javase/8/docs/api/>

**Python** <https://docs.python.org/3/>

**Kotlin** <https://kotlinlang.org/docs/reference/>

- 대회 종료 전에 퇴실할 수 없습니다.
- 책이나 개인이 준비한 인쇄된 참고자료를 열람할 수 있습니다. 참고자료의 양에는 제한이 없습니다.
- 대회 중에 휴대폰 및 전자기기는 사용할 수 없습니다.

## 문제 목록

문제지에 있는 문제가 총 8문제가 맞는지 확인하시기 바랍니다.

- A 3대 측정
- B 서강근육맨
- C 반전 요세푸스
- D 민트 초코
- E 할로윈의 양아치
- F 비밀번호 제작
- G 피보나치와 수열과 쿼리
- H 블랙홀

## 문제 A. 3대 측정

시간 제한 3 초  
메모리 제한 1024 MB

웨이트 트레이닝에서의 3대 측정은 스쿼트, 벤치프레스, 데드리프트의 중량을 측정하는 것이다. 하지만 세 명이 한 팀을 이루어 출전하는 전국 대학생 프로그래밍 대회(ICPC)의 참가자들은 다소 독특한 방법으로 3대를 측정하는데, 바로 팀원 각각의 실력을 수치로 나타내 주는 ‘코드포스 레이팅’을 비교하는 것이다.

웨이트 트레이닝계에는 3대 중량을 합쳐 500kg를 넘지 못하는 사람은 ‘언더아머’ 브랜드의 옷을 입지 못한다는 암묵적인 룰이 있으며, 이들을 단속하는 ‘언더아머 단속반’이 존재한다는 소문도 있다. Sogang ICPC Team은 이를 벤치마킹해 팀원 3명의 코드포스 레이팅의 합이  $K$  미만인 팀은 가입할 수 없는 **VIP 클럽**을 만들고자 한다.

하지만 이런 조건에서는 세 명 중 한 명의 레이팅이 현저히 낮더라도 나머지 두 명의 레이팅이 충분히 높다면 세 명이 모두 VIP 클럽에 가입할 수 있게 된다. 클럽에 가입하는 사람들은 모두 일정 수준 이상이어야겠다고 판단한 학회장 임지환은 **모든 팀원의 레이팅이  $L$  이상이고, 팀원 세 명의 레이팅의 합이  $K$  이상인 팀만**이 가입할 수 있게 하였다.

학회 일과 연합 일에 치여 사는 지환이는 VIP 클럽의 회원 관리까지 할 시간이 없다. 지환이를 위해 지원자 중 VIP 클럽에 가입할 수 있는 팀의 수를 구하고, VIP 회원들의 레이팅을 출력해 보자.

### 입력

첫째 줄에 정수  $N, K, L$ 이 주어진다.  $N$ 은 팀의 수,  $K$ 는 팀원 3명의 레이팅 합에 대한 클럽 가입 조건,  $L$ 은 개인 레이팅에 대한 클럽 가입 조건이다. ( $1 \leq N \leq 500\,000, 0 \leq K \leq 12\,000, 0 \leq L \leq 4\,000$ )

둘째 줄부터  $N$ 개 줄에 VIP 클럽에 신청한 팀의 팀원들의 레이팅 정보를 나타내는 정수  $x_1, x_2, x_3$ 이 한 줄에 한 팀씩 주어진다. ( $0 \leq x_1, x_2, x_3 \leq 4\,000$ )

### 출력

첫째 줄에 VIP 클럽에 가입이 가능한 팀의 수를 출력한다.

둘째 줄에 VIP 회원들의 레이팅을 입력받은 순서대로 공백으로 구분해 하나씩 출력한다.

### 입출력 예시

| 표준 입력(stdin)                                                                                          | 표준 출력(stdout)                                        |
|-------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 5 5000 1600<br>1621 1928 1809<br>2300 2300 1499<br>1805 1211 1699<br>1600 1700 1800<br>1792 1617 1830 | 3<br>1621 1928 1809 1600 1700 1800 1792<br>1617 1830 |

이 페이지는 공백입니다

## 문제 B. 서강근육맨

시간 제한 1 초  
메모리 제한 1024 MB

로니 콜먼 동영상을 보고 보디빌더가 되기로 결심한 향빈이는 PT 상담을 받으러 서강헬스클럽에 갔다. 향빈이가 서강헬스클럽을 선택한 이유는 PT를 받을 때 사용하는 운동기구를 회원이 선택할 수 있다는 점 때문이다. 하지만, 서강헬스클럽은 항상 사람이 많아서 PT를 한 번 받을 때 운동기구를 최대 두 개까지만 선택할 수 있다.

헬스장에 있는  $N$ 개의 운동기구를 한 번씩 사용해보고 싶은 향빈이는 PT를 받을 때마다 이전에 사용하지 않았던 운동기구를 선택하기로 계획을 세웠다. 그리고 비용을 절약하기 위해 PT를 받을 때 운동기구를 되도록이면 두 개를 사용하기로 했다. 예를 들어, 헬스장에 총 10개의 운동기구가 있을 경우 PT를 5번 받으면 모든 기구를 다 사용할 수 있다. 9개의 운동기구가 있는 경우에도 PT를 5번 받지만, 마지막 PT를 받을 때는 운동기구를 하나만 사용한다.

하지만 향빈이는 운동기구를 선택하다가 큰 고민에 빠졌다. 왜냐하면 운동기구마다 근손실이 일어나는 정도가 다르기 때문이다. 어떤 운동기구는 자극이 잘 안 와서 근손실이 적게 일어나는데, 어떤 운동기구는 자극이 잘 와서 근손실이 많이 일어난다. 근손실이 죽음보다 무서운 향빈이는 PT를 한 번 받을 때의 근손실 정도가  $M$ 을 넘지 않도록 하고 싶다. 이때,  $M$ 의 최솟값을 구해보자. 참고로, 운동기구를 두 개 사용해서 PT를 받을 때의 근손실 정도는 두 운동기구의 근손실 정도의 합이다.

### 입력

첫째 줄에 서강헬스클럽에 비치된 운동기구의 개수를 나타내는 정수  $N$ 이 주어진다. ( $1 \leq N \leq 10\,000$ )

둘째 줄에는 각 운동기구의 근손실 정도를 나타내는 정수  $t_1, t_2, \dots, t_N$ 가 주어진다. ( $0 \leq t_i \leq 10^{18}$ )

### 출력

$M$ 의 최솟값을 출력한다.

### 입출력 예시

| 표준 입력(stdin)   | 표준 출력(stdout) |
|----------------|---------------|
| 5<br>1 2 3 4 5 | 5             |

PT 첫째 날에 1과 4를 선택하고, 둘째 날에 2와 3을 선택하고, 마지막 날에 5를 선택하면  $M$ 은 5가 되며, 이때가  $M$ 이 최소일 때이다.

이 페이지는 공백입니다

## 문제 C. 반전 요세푸스

시간 제한 1 초  
메모리 제한 1024 MB

요세푸스 문제는 다음과 같다.

1번 사람 오른쪽에는 2번 사람이 앉아 있고, 2번 사람 오른쪽에는 3번 사람이 앉아 있고, 계속하여 같은 방식으로  $N$ 명의 사람들이 원을 이루며 앉아 있다.  $N$ 번 사람 오른쪽에는 1번 사람이 앉아 있다. 이제  $K(\leq N)$ 번 사람을 우선 제거하고, 이후 직전 제거된 사람의 오른쪽의  $K$ 번째 사람을 계속 제거해 나간다. 모든 사람이 제거되었을 때, 제거된 사람의 순서는 어떻게 될까?

이 문제의 답을  $(N, K)$ -요세푸스 순열이라고 하며,  $(7, 3)$ -요세푸스 순열은  $\langle 3, 6, 2, 7, 5, 1, 4 \rangle$ 가 된다.

하지만 한 방향으로만 계속 돌아가는 건 너무 지루하다. 따라서 요세푸스 문제에 재미를 더하기 위해  $M$ 명의 사람이 제거될 때마다 원을 돌리는 방향을 계속해서 바꾸려고 한다. 이렇게 정의된 새로운 문제의 답을  $(N, K, M)$ -반전 요세푸스 순열이라고 하며,  $(7, 3, 4)$ -반전 요세푸스 순열은  $\langle 3, 6, 2, 7, 1, 5, 4 \rangle$ 가 된다.

$N, K, M$ 이 주어질 때,  $(N, K, M)$ -반전 요세푸스 순열을 계산해 보자.

### 입력

첫째 줄에 정수  $N, K, M$ 이 주어진다. ( $1 \leq N \leq 5\,000, 1 \leq K, M \leq N$ )

### 출력

$(N, K, M)$ -반전 요세푸스 순열을 이루는 수들을 한 줄에 하나씩 순서대로 출력한다.

### 입출력 예시

| 표준 입력(stdin) | 표준 출력(stdout)                   |
|--------------|---------------------------------|
| 7 3 4        | 3<br>6<br>2<br>7<br>1<br>5<br>4 |

이 페이지는 공백입니다

## 문제 D. 민트 초코

시간 제한 1 초 (추가 시간 있음)  
메모리 제한 1024 MB

상원이는 아주 특별한 방법으로 디저트를 고른다.

상원이는 정수의 곱셈과 나눗셈으로만 이뤄진 임의의 수식을 적고, 그 결과가 정수이면 “민트 초코”를, 정수가 아닌 유리수이면 “치약”을 먹기로 했다.

상원이가 적은 수식이 주어졌을 때, 어떤 디저트를 먹게 될지 맞춰보자.

### 입력

첫째 줄에 수식을 이루는 수의 개수  $N$ 이 주어진다. ( $1 \leq N \leq 100\,000$ )

둘째 줄에 수식이 주어진다. 수식은 정수와 연산자(\*, /)가 공백으로 구분되어 주어진다. 수식은 정수와 연산자가 번갈아 주어지며, 항상 정수로 시작해서 정수로 끝난다. 수식을 이루는 모든 정수는  $-100\,000$  이상  $100\,000$  이하이다.

올바른 수식만 주어지고, 0으로 나누는 경우는 주어지지 않는다.

### 출력

상원이가 고른 디저트가 “민트 초코”인 경우 `mint chocolate`, “치약”인 경우 `toothpaste`를 출력한다.

### 입출력 예시

| 표준 입력(stdin)               | 표준 출력(stdout)  |
|----------------------------|----------------|
| 6<br>1 * 2 / 3 / 4 * 5 * 6 | mint chocolate |
| 6<br>1 * 2 / 3 / 4 / 5 * 6 | toothpaste     |

이 페이지는 공백입니다

## 문제 E. 할로윈의 양아치

시간 제한 1 초 (추가 시간 있음)  
메모리 제한 1024 MB

Trick or Treat!!

10월 31일 할로윈의 밤에는 거리의 여기저기서 아이들이 친구들과 모여 사탕을 받기 위해 돌아다닌다. 올해 할로윈에도 어김없이 많은 아이가 할로윈을 즐겼지만 단 한 사람, 일찍부터 잠에 빠진 스브러스는 할로윈 밤을 즐길 수가 없었다. 뒤늦게 일어나 사탕을 얻기 위해 혼자 돌아다녀 보지만 이미 사탕은 바닥나 하나도 얻을 수 없었다.

단단히 화가 난 스브러스는 거리를 돌아다니며 다른 아이들의 사탕을 빼앗기로 마음을 먹는다. 다른 아이들보다 몸집이 큰 스브러스에게 사탕을 빼앗는 건 어렵지 않다. 또한, 스브러스는 매우 공평한 사람이기 때문에 한 아이의 사탕을 뺏으면 그 아이 친구들의 사탕도 모조리 뺏어버린다. (친구의 친구는 친구다?!)

사탕을 빼앗긴 아이들은 거리에 주저앉아 울고  $K$ 명 이상의 아이들이 울기 시작하면 울음소리가 공명하여 온 집의 어른들이 거리로 나온다. 스브러스가 어른들에게 들키지 않고 최대한로 뺏을 수 있는 사탕의 양을 구하여라.

스브러스는 혼자 모든 집을 돌아다녔기 때문에 다른 아이들이 받은 사탕의 양을 모두 알고 있다. 또한, 모든 아이는 스브러스를 피해 갈 수 없다.

### 입력

첫째 줄에 정수  $N, M, K$ 가 주어진다.  $N$ 은 거리에 있는 아이들의 수,  $M$ 은 아이들의 친구 관계 수,  $K$ 는 울음소리가 공명하기 위한 최소 아이의 수이다. ( $1 \leq N \leq 30\,000, 0 \leq M \leq 100\,000, 1 \leq K \leq \min\{N, 3\,000\}$ )

둘째 줄에는 아이들이 받은 사탕의 수를 나타내는 정수  $c_1, c_2, \dots, c_N$ 이 주어진다. ( $1 \leq c_i \leq 10\,000$ )

셋째 줄부터  $M$ 개 줄에 걸쳐 각각의 줄에 정수  $a, b$ 가 주어진다. 이는  $a$ 와  $b$ 가 친구임을 의미한다. 같은 친구 관계가 두 번 주어지는 경우는 없다. ( $1 \leq a, b \leq N, a \neq b$ )

### 출력

스브러스가 어른들에게 들키지 않고 아이들로부터 뺏을 수 있는 최대 사탕의 수를 출력한다.

(뒷면에 입출력 예시가 있습니다.)

**입출력 예시**

| 표준 입력(stdin)                                                                 | 표준 출력(stdout) |
|------------------------------------------------------------------------------|---------------|
| 10 6 6<br>9 15 4 4 1 5 19 14 20 5<br>1 3<br>2 5<br>4 9<br>6 2<br>7 8<br>6 10 | 57            |
| 5 4 4<br>9 9 9 9 9<br>1 2<br>2 3<br>3 4<br>4 5                               | 0             |

## 문제 F. 비밀번호 제작

시간 제한 1 초 (추가 시간 있음)

메모리 제한 1024 MB

서강대학교 전산실에서 보안직원으로 일하는 향빈이는 한 통의 이메일을 받게 되었다. 이메일에는 서버 관리자 계정에 대한 비정상적인 로그인 시도가 감지되었다는 내용이 적혀 있었고, 첨부된 파일에는 지금까지 로그인 시도에 사용된 비밀번호 목록이 있었다. 참고로, 서버 관리자 계정의 비밀번호로는 0 이상  $N$  이하의 정수 중 하나를 사용할 수 있다.

두 비밀번호의 **안전 거리**는 이진법으로 표현한 두 비밀번호의 서로 다른 자리의 개수로 정의한다. 예를 들어 3을 이진법으로 표현하면 0011, 8을 이진법으로 표현하면 1000이 되고, 이때 서로 다른 자리의 개수는 3개이므로 3과 8의 안전 거리는 3이 된다.

어떤 비밀번호의 **안전도**는 지금까지 로그인 시도에 사용된 모든 비밀번호와의 안전 거리 중 최솟값으로 정의한다. 예를 들어 지금까지 로그인 시도에 사용된 비밀번호가 3과 4이라고 가정하면, 새로운 비밀번호 8에 대해 3과 8의 안전 거리는 3, 4와 8의 안전 거리는 2이므로 비밀번호 8의 안전도는 2가 된다.

향빈이는 해커가 비밀번호를 알아내기까지의 시간을 최대한 늦추기 위해 현재 사용 중인 관리자 계정 비밀번호의 안전도가 가장 높게끔 바꾸고 싶다. 이때, 안전도가 제일 높은 비밀번호의 안전도를 구하여라.

### 입력

첫째 줄에 관리자 계정 비밀번호의 최댓값을 나타내는 정수  $N$ 이 주어진다. ( $0 \leq N \leq 1\,000\,000$ )

둘째 줄에는 로그인 시도에 사용된 비밀번호의 개수를 나타내는 정수  $M$ 이 주어진다. ( $1 \leq M \leq 100\,000$ )

셋째 줄에는 로그인 시도에 사용된 비밀번호 값인 정수  $p_1, p_2, \dots, p_M$ 이 주어진다. ( $0 \leq p_i \leq N$ )

### 출력

안전도가 제일 높은 비밀번호의 안전도를 출력한다.

### 입출력 예시

| 표준 입력(stdin)   | 표준 출력(stdout) |
|----------------|---------------|
| 10<br>2<br>3 4 | 2             |

비밀번호를 8 또는 9로 설정했을 때 안전도가 2로 최대이다.

이 페이지는 공백입니다

## 문제 G. 피보나치와 수열과 쿼리

시간 제한 1 초 (추가 시간 있음)  
메모리 제한 1024 MB

피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 2번째 피보나치 수부터는 바로 앞 두 피보나치 수의 합이 된다. 이를 식으로 표현하면  $F_n = F_{n-2} + F_{n-1}$  ( $n \geq 2$ )이 된다.

$n$ 번째 피보나치 수 하나는 위와 같은 점화식을 통해 쉽게 구할 수 있지만, 이번 문제는 호락호락하지 않다. 모든 값이 0인 길이  $N$ 의 수열이 주어진다. 이때, 다음 쿼리를 수행한 후의 결과를 출력하는 프로그램을 작성해보자.

- 1  $r$ : 수열의  $l$ 번째 위치부터  $r$ 번째 위치까지에 각각  $F_1, F_2, \dots, F_{r-l+1}$ 을 더한다.

### 입력

첫째 줄에 수열의 크기  $N$ 이 주어진다. ( $1 \leq N \leq 1\,000\,000$ )

둘째 줄에 쿼리의 개수  $Q$ 가 주어진다. ( $1 \leq Q \leq 100\,000$ )

셋째 줄부터  $Q$ 개 줄에 걸쳐 쿼리에 대한 정보  $l, r$ 이 주어진다. ( $1 \leq l \leq r \leq N$ )

### 출력

모든 쿼리를 적용한 후, 수열의 모든 수를 공백으로 구분해 출력한다. 단, 수가 너무 커질 수 있으니 각각의 수를  $10^9 + 7$ 으로 나눈 나머지를 출력한다.

### 입출력 예시

| 표준 입력(stdin)                  | 표준 출력(stdout)            |
|-------------------------------|--------------------------|
| 10<br>3<br>1 10<br>1 5<br>5 6 | 2 2 4 6 11 9 13 21 34 55 |

이 페이지는 공백입니다

## 문제 H. 블랙홀

시간 제한 1 초 (추가 시간 있음)  
메모리 제한 1024 MB

평화로운 도시에 갑자기 블랙홀이 생겼다. 블랙홀은 점점 커지며 도시의 건물을 전부 집어삼킬 것이다. 블랙홀은 다음과 같은 특징이 있다.

- 블랙홀은 평면 위의  $N$ 개의 점을 꼭짓점으로 갖는 볼록다각형으로 나타낼 수 있다.  $N$ 개의 점 중 어느 세 점도 한 직선 위에 있지 않다.
- 블랙홀 경계를 포함하지 않는 내부에는 특이점이 하나 존재하며, 처음에 블랙홀을 이루는 점들은 블랙홀이 생긴 지  $t$ 초가 되는 시점에 이 특이점으로부터  $(t+1)$ 배 떨어져 있다.
- 블랙홀은 블랙홀의 내부와 경계에 있는 건물을 집어삼킨다.

도시에는  $M$ 개의 건물이 있으며, 각 건물은 평면 상의 점으로 나타낼 수 있다.

당신은 매 초마다 블랙홀이 지금까지 집어삼킨 건물의 수를 세면서 블랙홀이  $K$ 개 이상의 건물을 집어삼켰으면 대피하려고 한다. 특이점, 블랙홀, 건물의 정보가 주어졌을 때에 대피하게 되는 시간을 구해보자.

### 입력

첫째 줄에 정수  $N, M, K$ 가 주어진다. ( $3 \leq N \leq 100\,000, 1 \leq K \leq M \leq 100\,000$ )

둘째 줄에 블랙홀의 특이점의 좌표가 주어진다.

셋째 줄부터  $N$ 개 줄에 걸쳐 블랙홀의 꼭짓점들의 좌표가 반시계방향 순서로 주어진다.

다음  $M$ 개 줄에 걸쳐 각 건물의 좌표가 주어진다.

모든 점의 좌표는  $-10^9$  이상  $10^9$  이하의 정수이다.

### 출력

대피 시간을 정수로 출력한다.

(뒷면에 입출력 예시가 있습니다.)

## 입출력 예시

| 표준 입력(stdin)                                                              | 표준 출력(stdout) |
|---------------------------------------------------------------------------|---------------|
| 4 4 3<br>0 0<br>1 1<br>-1 1<br>-1 -1<br>1 -1<br>2 0<br>3 0<br>4 4<br>-5 5 | 3             |

