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 를 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. 를 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.

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/