

ACM-ICPC 참가자들을 위한 도움말

2011년 9월 27일, 버전 1.0.2

algotspot.com 운영진 씬

참가자들을 위한 도움말

이 글은 주로 대회에 처음 참가하는 사람들을 위해 [algospot](http://www.algospot.com/)(<http://www.algospot.com/>)의 운영진들이 작성하였습니다. 다소 긴 내용이더라도 반드시 유념하여 대회에서 실력 이외의 이유로 불이익을 받지 않길 바라며 모쪼록 참가자 여러분이 대회의 참된 재미를 느끼시길 기대합니다.

채점 결과 요약

Yes 제출하신 답안이 모든 테스트 데이터를 정해진 시간 안에 통과하여 정답으로 인정되었음을 의미합니다.

No - Compilation Error 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

No - Run-time Error 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

No - Time-limit Exceeded 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

No - Wrong Answer 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

No - Excessive Output 제출하신 답안 프로그램이 지나치게 많은 양을 출력하여 강제로 종료되었음을 의미합니다.

No - Output Format Error 제출하신 답안 프로그램이 정해진 출력 형식을 따르지 않았음을 의미합니다.

No - Other - Contact Staff 위에서 나열한 이유 이외의 문제로 채점이 거부되었음을 의미하며, 설명 요청 기능이나 기타 통신 수단을 통해 대회 본부에 문의해 주십시오.

위에서 언급한 모든 채점 결과에 있어 둘 이상의 오류가 있는 경우 그 중 어느 것도 결과로 결정될 수 있음을 유념하십시오.

입력과 출력

프로그램의 모든 입력은 **표준 입력**(Standard Input)에 주어지며, 모든 출력은 **표준 출력**(Standard Output)을 통해 이루어져야 합니다.

이 때 채점은 주로 **심판**(Judge)이 준비한 채점 데이터 파일을 리디렉션(Redirection)하여 이루어지는데, 이 탓에 임의로 채점이 사람의 손에 의해 입력될 것이라는 가정을 해선 안됩니다. 많은 경우 자동화된 도구를 통해 처리합니다.

팁: 모든 입력이 주어진 다음에 출력을 처리할 필요가 없으며, 입력을 받는 대로 처리한다고 생각하면 됩니다. 이 글 뒷 부분의 예시 코드를 참조합니다.

주의: `getch()` 등의 콘솔과 관련된 함수를 호출해서는 안됩니다.

주의: 특히 C에서 `fflush(stdin)`은 사용해서는 안됩니다. 아래에서 좀 더 자세히 다룹니다.

fflush의 입력 파일에 대한 동작은 표준에서 정의되지 않은 것일 뿐더러, 설령 기대대로 버퍼를 비우는 동작을 한다고 하면 채점용 테스트 데이터 파일에서 일정 단위로 읽어 채운 버퍼를 비워 버리게 되며, 이 경우 프로그램이 어떻게 동작할지는 전혀 알 수 없게 됩니다. 따라서 절대 사용해서는 안 되며, 행 끝의 줄 바꿈 문자 등이 문제가 되는 경우 다른 방법으로 잘 처리해야 합니다.

주의: 줄 바꿈 문자가 DOS(CR LF, 0x0D 0x0A), UNIX(LF, 0x0A), MAC(CR, 0x0D) 스타일 중 어떤 것을 따르는지는 상황에 따라 달라질 수 있으므로 임의로 가정해서는 안 됩니다.

주의: gets()의 경우 마지막에 줄 바꿈 문자가 상황에 따라 주어지기도 하고 주어지지 않기도 하므로, 방어적으로 처리하거나 fgets()와 같은 대체 함수를 활용합니다.

주의: 입력의 맨 끝에 빈 줄이 주어진다는 가정은 하지 않는 것이 좋습니다.

심판은 프로그램이 내어 놓은 출력을 정답과 비교하여 검증하는데, 이 또한 대부분 자동화된 도구를 통해 처리합니다. 심판은 정답이 아닌 경우 “No - Wrong Answer”를 결과로 줍니다.

주의: 출력은 문제에서 지정한 것과 토씨 하나, 경우에 따라서는 공백의 개수 하나도 틀려서는 안 됩니다. 특히 요구된 대로 줄바꿈하는지 반드시 확인합니다.

주의: 상황에 따라 다를 수 있으나 일반적으로 줄 끝에는 공백이 있어서는 안 됩니다. 특히 여러 개의 값을 공백으로 구분해서 출력할 때 유의해야 합니다.

주의: 디버그를 위해 임시로 출력하는 것들은 제출할 때 반드시 제거해야 합니다.

흔치 않지만, 심판은 상황에 따라 지나치게 많은 출력을 내어놓는 경우 채점 도중 중단시키고 “No - Excessive Output”를 결과로 주기도 하며, 출력 양식이 올바르지 않으면 “No - Output Format Error”를 결과로 주기도 합니다.

컴파일

참가자는 반드시 소스 코드를 제출해야 합니다.

주의: exe, class, vcproj, sln 파일 등을 제출해서는 안 됩니다.

많은 경우 채점 시스템은 하나의 파일만을 제출 가능하도록 합니다. PC²의 경우 여러 개의 파일을 제출할 수 있으나 권장되지 않습니다. 가급적 하나의 파일로 컴파일 가능하도록 합니다.

팁: Visual Studio 사용자의 경우 Win32 콘솔 어플리케이션을 생성하되, 빈 프로젝트 옵션을 선택하면 원하는 결과를 얻을 수 있습니다.

주의: 컴파일 설정은 심판이 미리 지정하므로, 별도의 컴파일 옵션이 필요하도록 프로그램을 작성해서는 안 됩니다.

컴파일과 그 이후의 과정은 심판진들에게서 이루어지므로, 반드시 사전에 지정된 컴파일러 버전을 사용해야 합니다.

주의: 같은 컴파일러라고 하더라도 버전에 따라 많은 차이가 있을 수 있으니 다른 버전의 사용을 자제합니다. C/C++의 경우 헤더 의존 관계가 변경된다거나 전역 식별자(identifier)가 변경될 수 있습니다.

주의: Microsoft Visual C++ 6.0은 절대로 사용하지 않습니다.

또한, 채점 시스템이 파일 이름에 제약을 가하기도 합니다.

주의: 파일 이름에 절대로 공백, 특수 문자, ASCII 문자 셋에 포함되지 않은 기타 한글 등을 쓰지 않아야 합니다.

주의: Java 사용자의 경우 언어의 규칙을 따라 소스 코드의 이름과 클래스 이름이 같아야 합니다.

어떤 이유에서든, 답안이 심판의 시스템에서 정상적으로 컴파일되지 않는다면 “No - Compilation Error” 를 받게 됩니다.

팁: PC²의 경우 클라이언트에 Test 기능이 내장되어 있는데, 이는 심판진들이 설정한 환경과 동일한 경우 컴파일을 해 볼 수 있는 기능이므로 신경쓰지 않으셔도 됩니다. 채점이나 제출과는 전혀 관련이 없습니다.

프로그램의 수행과 종료

프로그램은 반드시 정해진 시간 내에 종료되어야 합니다. 만일 시간 내에 종료되지 않으면 “No - Time-limit Exceeded” 를 받게 됩니다. 따라서, 주어진 입력 크기를 처리할 수 있는 충분히 빠른 알고리즘을 고안해야 합니다. 출제, 입력의 범위 제한, 시간 제한은 어떤 수준의 알고리즘을 맞게 처리할 것인지를 깊게 논의한 끝에 정해집니다.

팁: 대체로, 1억 번 가량의 단위 연산에 1초정도의 시간이 걸린다고 짐작하면 됩니다. 예를 들어 단순한 $O(n^2)$ 알고리즘은 $n \leq 5,000$ 정도라면 맞을 확률이 높습니다.

팁: 문제에서 따로 언급하지 않은 경우 시간 제한은 참가자에게 공개되지 않은 것입니다.

한편 프로그램이 수행 도중 비정상적으로 종료되는 경우 “No - Run-time Error” 를 받게 됩니다. 여기에는 프로그램이 처리하지 않은 예외(Exception), 0으로 나누기, 잘못된 메모리 접근, 순수 가상 함수(Pure virtual function)의 호출 등 여러 가지 이유가 있을 수 있습니다.

팁: 심판은 어떤 원인으로 프로그램이 종료되었는지 알려주지 않습니다.

주의: 심판은 많은 경우 출력이 정답이더라도 프로그램이 0이 아닌 리턴 코드를 되돌릴 경우 잘못된 종료로 간주합니다.

주의: 따라서 C/C++에서 main()의 리턴 타입은 반드시 int여야 하며, 0을 되돌려야 합니다.

답안의 제출과 설명 요청

답안의 작성이 종료되어 제출하고자 한다면 컨테스트 시스템을 사용해 제출하면 되며, 크게 어려운 부분은 없습니다. 제출에 성공하면 시스템으로부터 성공적으로 제출되었음을 알리는 응답이 오며, 이 응답이 오지 않은 경우는 서버가 다른 응답을 처리하느라 바쁘거나 네트워크 연결이 좋지 않은 상태이므로 다시 제출할 지 아닐 지를 판단해야 합니다.

심판은 앞에서 언급한 모든 부분을 확인하여 결과를 정하며 시스템이 그 결과를 해당 팀에게 알려줍니다.

팁: 심판의 응답은 대회 진행 상황과 쌓인 제출의 수에 따라 다소 늦어질 수 있습니다.

문제의 지문에 이해가 잘 가지 않는 부분이 있다면 **설명**(Clarification)을 요청할 수 있습니다.

주의: 설명을 요청할 때 서로 간의 예절을 지켜 바른 어휘를 사용하시기 바랍니다. 가는 말이 고와야 오는 말이 곱습니다.

주의: 많은 수의 설명 요청은 기본 응답인 “No response, read problem statement carefully”로 되돌려지기 때문에 특별히 문제에 누락된 부분이 있는 것 같거나 모호한 부분이 있는 경우 이외에는 사용하지 않아야 합니다.

이하는 거의 대답해드리지 않는 질문의 예시입니다.

- 예제 데이터가 잘못된 것 같습니다.
- 입력이 이렇게 주어지면 답이 얼마인가요?
- 테스트 케이스의 수는 몇 개나 되나요? — 문제에서 주어진 경우 문제를 읽으시고, 문제에서 주어지지 않은 경우 알려드리지 않습니다.
- 시간제한은 몇 초인가요? — 위와 같습니다.
- 제 솔루션이 왜 WA 인가요? 제 시스템에서는 맞게 결과가 나옵니다.
- 제 시스템에서는 컴파일러가 잘 되는데 왜 “No - Compilation Error” 를 받나요?

순위표

보통 참가자들에게 간략한 정보를 담은 **순위표**(Scoreboard)를 제공합니다. 순위표는 거의 실시간으로 갱신 되지만, 대회 종료를 일정 시간 앞두고 갱신을 중단시킵니다. 이는 대회의 극적 긴장감을 고조시키기 위한 장치입니다.

순위표에는 보통 ACM-ICPC의 규칙을 따라 풀 문제 수, 팀의 고유번호, 페널티 등이 주어지며 심판진의 결정에 따라서 풀 문제를 나타내는 색상이나 문제 번호, 제출 횟수 등이 포함될 수 있습니다.

기타 주의할 대회 규칙

주의: 대회는 1팀 1PC를 사용하도록 강제하고 있으며, 반드시 지켜셔야 합니다.

주의: 대회 시스템 접근 이외의 용도로 인터넷을 사용하는 것 역시 금지되어 있습니다.

주의: bison, yacc 등의 자동 코드 생성 도구를 사용하는 것 또한 금지사항입니다.

코드 예시

다음은 각 언어별 표준 입/출력을 통해 테스트 케이스의 수를 입력받고 각 테스트 케이스에 대해 두 정수를 입력받아 합을 출력하는 프로그램입니다.

C

```
#include <stdio.h>

int main()
{
5     int t;
    scanf("%d", &t);
    for (int i = 0; i < t; i++)
    {
10         int a, b;
        scanf("%d %d", &a, &b);
        printf("%d\n", a + b);
    }
    return 0;
}
```

C++

```
1 #include <iostream>

using namespace std;

int main()
6 {
    int t;
    cin >> t;
    for (int i = 0; i < t; i++)
    {
11         int a, b;
        cin >> a >> b;
        cout << a + b << endl;
    }
    return 0;
16 }
```

Java

```
import java.util.*;
```

```
public class Adder
4 {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();
9     for (int i = 0; i < t; i++)
        {
            int a = sc.nextInt();
            int b = sc.nextInt();
            System.out.println(a + b);
14        }
    }
}
```