

#01-2. 입출력

나정휘

<https://justicehui.github.io/>

목차

온라인 저지의 채점 방식

EOF 처리

다양한 입출력 형식의 처리

온라인 저지의 채점 방식

온라인 저지의 채점 방식

일반적인 채점 방식

- 각 문제마다 한 쌍 이상의 (입력 파일, 출력 파일) 쌍이 있음
 - 채점 서버에는 예제 외에도 더 많은 데이터가 있음
- 프로그램에 xx.in을 넣어서 나온 결과가 xx.out과 같은지 확인
- 모든 파일에서 동일한 결과가 나와야 맞았습니다!! 를 받음
- 프로그램의 구현 방식은 신경 쓰지 않음
- 제한 시간 안에 메모리 제한을 넘기지 않으면서
- 올바른 정답을 출력하고 종료하기만 하면 됨

2656번 - 선린인터넷고등학교 교가

[데이터 전체 삭제](#)[데이터 추가하기](#) i

모든 입/출력 파일의 앞/뒤 공백은 제거되고, 줄 바꿈도 (\n)으로 바뀝니다. (요약: `lines = [line.strip()+'\n' for line in fp]`)
공백이 유지되어야 하는 경우에는 관리자에게 문의해주세요.

파일 이름	크기	상태	다운로드	삭제
01.in	16 B	완료	다운로드	
01.out	6 B	완료	다운로드	
02.in	8 B	완료	다운로드	
02.out	6 B	완료	다운로드	
03.in	105 B	완료	다운로드	
03.out	6 B	완료	다운로드	
04.in	505 B	완료	다운로드	
04.out	6 B	완료	다운로드	
05.in	1006 B	완료	다운로드	
05.out	6 B	완료	다운로드	

온라인 저지의 채점 방식

BOJ 1000. A+B

- 두 방법 모두 괜찮음

```

//
#include <stdio.h>

int main(){
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d", a + b);
}

////////////////////////////////////

#include <stdio.h>

int ans[10][10] = {
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9},
    {1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
    {2, 3, 4, 5, 6, 7, 8, 9, 10, 11},
    {3, 4, 5, 6, 7, 8, 9, 10, 11, 12},
    {4, 5, 6, 7, 8, 9, 10, 11, 12, 13},
    {5, 6, 7, 8, 9, 10, 11, 12, 13, 14},
    {6, 7, 8, 9, 10, 11, 12, 13, 14, 15},
    {7, 8, 9, 10, 11, 12, 13, 14, 15, 16},
    {8, 9, 10, 11, 12, 13, 14, 15, 16, 17},
    {9, 10, 11, 12, 13, 14, 15, 16, 17, 18},
};

int main(){
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d", ans[a][b]);
}

```

온라인 저지의 채점 방식

BOJ 10950. A+B - 3

- 첫째 줄에 테스트 케이스의 개수 T가 주어지고
- 이후 T줄에 각각 A, B가 주어지면 A+B를 출력하는 문제
- 단순히 프로그램의 출력과 xx.out의 내용만 비교하기 때문에
- 한 번에 모아서 출력하지 않고 입력 들어올 때마다 바로바로 출력해도 됨

```
#include <stdio.h>

int main(){
    int T;
    scanf("%d", &T);
    for(int tc=1; tc<=T; tc++){
        int a, b;
        scanf("%d %d", &a, &b);
        printf("%d\n", a + b);
    }
}
```

온라인 저지의 채점 방식

스페셜 저지

- 정답이 여러 가지인 문제는 출력값이 정답인지 판정하는 채점 프로그램을 이용해 채점

BOJ 1008. A/B

- 두 정수 A, B가 주어지면 A/B를 출력하는 문제
- 실제 정답과 출력값의 절대오차 또는 상대오차가 10^{-9} 이하이면 정답 처리
 - 절대오차 = $|참값 - 측정값|$
 - 상대오차 = 절대오차 / 참값
 - 실제 정답이 $x(> 0)$ 면 $\min\{x-10^{-9}, (1-10^{-9})x\}$ 이상 $\max\{x+10^{-9}, (1+10^{-9})x\}$ 이하를 출력하면 됨
 - 만약 실제 정답이 1.0 이면 0.999999999999 를 출력해도 정답 처리한다는 뜻
 - <https://github.com/MikeMirzayanov/testlib/blob/master/checkers/rcmp9.cpp> 참고
- `int a, b; scanf("%d %d", &a, &b);`
- `printf("%.10lf", (double)a / b);` 또는 `printf("%.10lf", 1.0 * a / b);`

질문?

형변환 꿀팁

캐스팅 연산자

- (type-name)cast-expression → cast-expression의 값을 type-name형으로 변환
 - (double)(a + b)는 a + b의 결과를 double형로 변환

리터럴

- 10, 10L, 10LL은 각각 int, long, long long (소문자 l도 가능)
 - unsigned로 만들고 싶으면 10u, 10ul, 10ull처럼 쓰면 됨
 - https://en.cppreference.com/w/cpp/language/integer_literal
- 1.0f는 float, 1.0은 double (대문자 F도 가능)
 - 소수점 아래는 생략해도 됨 (ex. 1.F)
 - 5e4, 7e-9 같은 것도 가능
 - $5e4 = 5 * 10^4$, $7e-9 = 7 * 10^{-9}$, f 붙이면 float
 - https://en.cppreference.com/w/cpp/language/floating_literal
- 캐스팅 연산자 쓰기 귀찮으면 1LL, 1.0 곱하거나 0LL, 0.0 더하면 됨

형변환 꿀팁



```
#include <stdio.h>
```

```
int main(){  
    int a, b;  
    a = b = 100'000;  
    printf("%lld %lld\n", (long long)a * b, 1LL * a * b);  
    // 10000000000 10000000000  
    a = b = 2'000'000'000;  
    printf("%lld %lld\n", (long long)a + b, 0LL + a + b);  
    // 4000000000 4000000000  
    a = 10; b = 3;  
    printf("%lf %lf\n", (double)a / b, 1.0 * a / b);  
    // 3.333333 3.333333  
}
```

EOF 처리

EOF 처리

BOJ 10951. A+B - 4

- 입력은 여러 개의 테스트 케이스로 이루어져 있다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다.
- A+B - 3 과 다른 점
 - 테스트 케이스의 개수가 주어지지 않음
 - 언제까지 입력을 받아야 할까? → 입력 파일의 끝(end of file)까지

EOF 처리

BOJ 10951. A+B - 4

- 입력은 여러 개의 테스트 케이스로 이루어져 있다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다.
- 파일의 끝(end of file)까지 입력받아야 하는 문제
- scanf의 반환 값
 - 성공적으로 입력받아서 저장한 인자의 개수
 - ex) scanf("%d %d", &a, &b);
 - 10 10 입력하면 2 반환, 10 a 입력하면 1 반환, a b 입력하면 0 반환
 - 만약 첫 번째 인자를 입력받기 전에 EOF를 만나면 음수 반환
 - 대부분 #define EOF (-1) 로 정의되어 있어서 -1을 반환함
 - 언어 레퍼런스에는 음수라고만 나와 있음
 - <https://en.cppreference.com/w/c/io> 참고

```
#include <stdio.h>

int main(){
    int a, b;
    while(scanf("%d %d", &a, &b) != EOF){
        printf("%d\n", a + b);
    }
}
```

질문?

다양한 입출력 형식의 처리

다양한 입출력 형식의 처리

BOJ 10952. A+B - 5

- 입력은 여러 개의 테스트 케이스로 이루어져 있다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다.
- 입력의 마지막에는 0 두 개가 들어온다.

```

#include <stdio.h>

int main(){
    int a, b;
    while(1){
        scanf("%d %d", &a, &b);
        if(a == 0 && b == 0) break;
        else printf("%d\n", a + b);
    }
}

////////////////////////////////////

#include <stdio.h>

int main(){
    int a, b;
    while(scanf("%d %d", &a, &b) && a && b){
        printf("%d\n", a + b);
    }
}
```


다양한 입출력 형식의 처리

BOJ 10953. A+B - 6

- 첫째 줄에 테스트 케이스의 개수 T가 주어진다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다. ($0 < A, B < 10$)
- A와 B는 콤마(,)로 구분되어 있다.
 - $T \geq 0$ 이면 while(T--) 는 정확히 T번 반복함

```
#include <stdio.h>

int main(){
    int T;
    scanf("%d", &T);
    while(T--){
        int a, b;
        scanf("%d,%d", &a, &b);
        printf("%d\n", a + b);
    }
}
```

```
#include <stdio.h>

int main(){
    int T;
    scanf("%d", &T);
    while(T--){
        char s[4];
        scanf("%s", s);
        int a = s[0] - '0', b = s[2] - '0';
        printf("%d\n", a + b);
    }
}
```

다양한 입출력 형식의 처리

BOJ 10953. A+B - 6

- 첫째 줄에 테스트 케이스의 개수 T가 주어진다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다. ($0 < A, B < 10$)
- A와 B는 콤마(,)로 구분되어 있다.
- A, B ≥ 10 이면 문자열 \rightarrow 정수 변환을 구현하면 됨
 - $123 = 12 * 10 + 3 = (1 * 10 + 2) * 10 + 3$ 임을 이용
 - 물론 `scanf("%d,%d", &a, &b)`도 여전히 가능

```
#include <stdio.h>
#include <string.h>

int main(){
    int T;
    scanf("%d", &T);
    while(T--){
        char s[11];
        scanf("%s", s);
        int len = strlen(s), pos = -1;
        for(int i=0; i<len; i++){
            if(s[i] == ','){
                pos = i;
                break;
            }
        }
        int a = 0, b = 0;
        for(int i=0; i<pos; i++) a = a * 10 + s[i] - '0';
        for(int i=pos+1; i<len; i++) b = b * 10 + s[i] - '0';
        printf("%d\n", a + b);
    }
}
```

다양한 입출력 형식의 처리

BOJ 10953. A+B - 6

- 첫째 줄에 테스트 케이스의 개수 T가 주어진다.
- 각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다. ($0 < A, B < 10$)
- A와 B는 콤마(,)로 구분되어 있다.
- 정수 → 문자열 변환도 구현한 코드
 - 수의 길이를 구한 다음
 - 일의 자리부터 차례대로 구함
 - 일의 자리 = $x \% 10$
 - 뒤 1자리 제거 = $x / 10$



```
#include <stdio.h>
#include <string.h>

int main(){
    int T;
    scanf("%d", &T);
    while(T--){
        char s[11], out[11] = {0};
        scanf("%s", s);
        int len = strlen(s), pos = -1;
        for(int i=0; i<len; i++){
            if(s[i] == ','){
                pos = i;
                break;
            }
        }
        int a = 0, b = 0;
        for(int i=0; i<pos; i++) a = a * 10 + s[i] - '0';
        for(int i=pos+1; i<len; i++) b = b * 10 + s[i] - '0';
        int size = 0;
        for(int i=a+b; i; i/=10) size++;
        for(int i=a+b; i; i/=10) out[--size] = i % 10 + '0';
        printf("%s\n", out);
    }
}
```

질문?