



# D.COM ALGORITHM STUDY

2020 - 07 VER.

1강) 기초 중의 기초 입출력!

# 1강. 기초 중의 기초 입력과 출력!

# 오늘의 목표

우선, 알고리즘 문제 해결 과정에 대해 알아보시다!

그리고 알고리즘 문제에 등장하는  
입출력 유형들을 정리해봅시다!

다 풀어놓고 입출력 잘못하면 맴찝...

#입출력 꿀팁도 알려드립니다!

가볍게,  
읽을거리



링크

## 지난 5년동안 소프트웨어 개발자로서 내가 배운 것들 [번역]

“자기 자신을 깊은 곳까지 밀어 넣어라, 타인의 얘기를 잘 듣는 열린 자세를 가져라.  
하지만 자기주장을 펴는 걸 두려워하지는 마라,  
변화를 받아들여라, 하지만 시류에 편승하지는 마라, 일과 놀이의 균형을 맞춰라”

본격적으로 알고리즘 문제들을 풀기 전에,

알고리즘 문제는 **어떻게** 생겼을까요?

또한 어떻게 **접근**하면 좋을까요?

# 알고리즘 문제 구성

크게  
문제 설명  
요구 조건  
입력  
출력  
예제  
로 이루어져 있습니다

연속합

성공

☆

| 시간 제한 | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
|-------|--------|-------|-------|-------|---------|
| 2 초   | 128 MB | 59152 | 16579 | 11361 | 27.285% |

## 문제

$n$ 개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다.  
예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.

## 입력

첫째 줄에 정수  $n$  ( $1 \leq n \leq 100,000$ )이 주어지고 둘째 줄에는  $n$ 개의 정수로 이루어진 수열이 주어진다. 수는 -1,000보다 크거나 같고, 1,000보다 작거나 같은 정수이다.

## 출력

첫째 줄에 답을 출력한다.

### 예제 입력 1 복사

```
10
10 -4 3 1 5 6 -35 12 21 -1
```

### 예제 출력 1 복사

```
33
```

# 알고리즘 문제 구성

정보올림피아드  
기업 코딩테스트  
ACM-ICPC 등  
대부분의 알고리즘 문제는  
다음과 같은 구성입니다.

연속합

성공

☆

| 시간 제한 | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
|-------|--------|-------|-------|-------|---------|
| 2 초   | 128 MB | 59152 | 16579 | 11361 | 27.285% |

## 문제

$n$ 개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다.

예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.

## 입력

첫째 줄에 정수  $n$  ( $1 \leq n \leq 100,000$ )이 주어지고 둘째 줄에는  $n$ 개의 정수로 이루어진 수열이 주어진다. 수는 -1,000보다 크거나 같고, 1,000보다 작거나 같은 정수이다.

## 출력

첫째 줄에 답을 출력한다.

### 예제 입력 1 복사

```
10
10 -4 3 1 5 6 -35 12 21 -1
```

### 예제 출력 1 복사

```
33
```

# 알고리즘 문제 접근방식

무작정 문제를 풀려고 하기 보다는  
자신에게 맞는 문제 접근 방식을 습득하는 것이  
중요합니다.



# 1단계 문제를 읽고 이해하기

당연하지만,  
우선 문제를 읽고 이해해야 합니다.  
초보뿐만 아니라 고수들도 문제를 잘못 읽는 실수를 범하곤 합니다.  
따라서 이 과정에서 꼼꼼하고 완벽하게 문제를 이해해야 합니다.

이 과정에서 자신이 다루기 쉬운 개념을 이용해서  
문제를 자신만의 언어로 풀어 쓰면 도움이 됩니다. (재정의와 추상화)

| 연속합 <span>성공</span> <span>☆</span>   |        |       |       |       |         |
|--|--------|-------|-------|-------|---------|
| 시간 제한  | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
| 2 초  | 128 MB | 59152 | 16579 | 11361 | 27.285% |
| 문제   |        |       |       |       |         |
| n개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다. |        |       |       |       |         |
| 예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.                |        |       |       |       |         |

# 2단계 계획 세우기

| 연속합 <span>성공</span> <span>☆</span>   |        |       |       |       |         |
|--|--------|-------|-------|-------|---------|
| 시간 제한  | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
| 2 초  | 128 MB | 59152 | 16579 | 11361 | 27.285% |
| 문제   |        |       |       |       |         |
| n개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다. |        |       |       |       |         |
| 예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.                |        |       |       |       |         |

이제 문제를 이해했다면, 해결 전략을 세워야 합니다.  
문제를 어떤 방식으로 해결할지,  
어떤 알고리즘과 자료 구조를 사용해야 하는지 선택해야 합니다.  
따라서 이번 스터디에서는 코딩 테스트에서  
주로 사용되는 알고리즘과 자료 구조를 배우게 됩니다.

# 3단계 계획 검증하기

| 시간 제한  | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
|--|--------|-------|-------|-------|---------|
| 2 초  | 128 MB | 59152 | 16579 | 11361 | 27.285% |
| <b>문제</b>  |        |       |       |       |         |
| <p>n개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다.</p> <p>예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.</p> |        |       |       |       |         |
| <b>입력</b>  |        |       |       |       |         |
| <p>첫째 줄에 정수 n(<math>1 \leq n \leq 100,000</math>)이 주어지고 둘째 줄에는 n개의 정수로 이루어진 수열이 주어진다. 수는 -1,000보다 크거나 같고, 1,000보다 작거나 같은 정수이다.</p>   |        |       |       |       |         |

자신이 계획한 방법이 맞는지 검증해보아야 합니다.  
과연 내가 세운 알고리즘이 정확한 답을 내놓는지,  
시간 제한, 메모리 제한, 입력 범위 안에서 올바른 작동을 하는지 확인해야 합니다.  
이를 위해 추후 강의 자료에서 알고리즘의 효율성과 정당성을  
분석하는 방법에 대해 배웁니다.

# 4단계 계획 수행하기

| 시간 제한  | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
|--|--------|-------|-------|-------|---------|
| 2 초  | 128 MB | 59152 | 16579 | 11361 | 27.285% |
| <b>문제</b>  |        |       |       |       |         |
| <p>n개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다.</p> <p>예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.</p> |        |       |       |       |         |
| <b>입력</b>  |        |       |       |       |         |
| <p>첫째 줄에 정수 <math>n</math> (<math>1 \leq n \leq 100,000</math>)이 주어지고 둘째 줄에는 n개의 정수로 이루어진 수열이 주어진다. 수는 -1,000보다 크거나 같고, 1,000보다 작거나 같은 정수이다.</p>   |        |       |       |       |         |

문제 풀이 계획을 무사히 수립했다면, 이제 코드로 옮길 시간입니다.  
아무리 완벽한 계획을 세웠다고 해도 정확하게 코드로 구현하지 못하면 말짱 도루묵입니다.  
이 과정에서 정확하고 빠르게 코드로 구현하기 위한 개발 가이드라인과 패턴 또한  
앞으로의 강의 자료에서 계속해서 다룹니다.

# 5단계 회고하기

| 시간 제한 | 메모리 제한 | 제출    | 정답    | 맞은 사람 | 정답 비율   |
|-------|--------|-------|-------|-------|---------|
| 2 초   | 128 MB | 59152 | 16579 | 11361 | 27.285% |

## 문제

n개의 정수로 이루어진 임의의 수열이 주어진다. 우리는 이 중 연속된 몇 개의 수를 선택해서 구할 수 있는 합 중 가장 큰 합을 구하려고 한다. 단, 수는 한 개 이상 선택해야 한다.

예를 들어서 10, -4, 3, 1, 5, 6, -35, 12, 21, -1 이라는 수열이 주어졌다고 하자. 여기서 정답은 12+21인 33이 정답이 된다.

## 입력

첫째 줄에 정수  $n$  ( $1 \leq n \leq 100,000$ )이 주어지고 둘째 줄에는 n개의 정수로 이루어진 수열이 주어진다. 수는 -1,000보다 크거나 같고, 1,000보다 작거나 같은 정수이다.

가장 중요한 단계입니다.

문제를 맞추는 그 자체보다, 자신의 실력을 향상하는 것이 더더욱 중요합니다.

이를 위해 코드에 자신의 생각을 기록하는 방법을 추천 드립니다.

문제를 다 풀고, 주석으로 접근 방식, 해결 아이디어, 어려웠던 점, 오답 원인 등을 기록해 보시기 바랍니다. 다음에 더욱 효율적이고 간결한 알고리즘을 찾거나 자주 실수하는 부분을 고쳐 나갈 수 있습니다.

또한 다른 사람의 코드를 보며 자신이 생각하지 못했던 방법을 습득해보는 것도 좋은 방법입니다.

자, 그럼 **입출력**을 한번 배워봅시다.

앞으로 수 많은 알고리즘 문제를 풀며 접하게 될,

대표적인 **입출력** 유형에는 **4**가지가 있습니다.

1.입력 데이터의 개수가 주어진 경우.

2.테스트 케이스의 개수가 주어진 경우.

3.입력의 개수를 모르는 경우.

4.입력을 한 줄 씩 받아야 하는 경우.



## 1. 입력 데이터의 개수가 주어진 경우.

어려운 것 없습니다.  
늘 하던 대로, 문제에 맞게 입 출력하면 됩니다.  
단, 문제의 조건을 꼼꼼히 확인하고  
변수의 자료형에 유의해주세요.

문제

두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 A와 B가 주어진다. ( $0 < A, B < 10$ )

출력

첫째 줄에 A+B를 출력한다.

예제 입력 1 복사

1 2

예제 출력 1 복사

3

## 1. 입력 데이터의 개수가 주어진 경우.



```
cin >> a >> b;  
cout << a+b << endl;
```



```
scanf("%d %d",&a,&b);  
printf("%d\n",a+b);
```

### 문제

두 정수 A와 B를 입력받은 다음, A+B를 출력하는 프로그램을 작성하시오.

### 입력

첫째 줄에 A와 B가 주어진다. ( $0 < A, B < 10$ )

### 출력

첫째 줄에 A+B를 출력한다.

### 예제 입력 1 복사

1 2

### 예제 출력 1 복사

3

Q. scanf printf 함수는 처음보는데요?  
A. 두 함수는 강의자료 뒤편에서 다루겠습니다.



연습 문제  
1000번 A+B

<https://www.acmicpc.net/problem/1000>

## 2. 테스트 케이스의 개수가 주어진 경우

각 테스트케이스를 독립적인 문제라고  
생각하시면 됩니다.  
즉, 한번에 모든 테스트케이스를 저장하고  
정답을 출력하지 않아도 됩니다.  
테스트케이스 하나 받고 정답 하나 출력하면 됩니다.

다음 슬라이드에서 코드로 알아보시다.

입력

첫째 줄에 테스트 케이스의 개수  $T$ 가 주어진다.

각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에  $A$ 와  $B$ 가 주어진다. ( $0 < A, B < 10$ )

출력

각 테스트 케이스마다  $A+B$ 를 출력한다.

예제 입력 1 복사

5  
1 1  
2 3  
3 4  
9 8  
5 2

예제 출력 1 복사

2  
5  
7  
17  
7

## 2. 테스트 케이스의 개수가 주어진 경우

```
int t;
int a[100], b[100];
cin >> t;
for (int i = 0; i < t; i++)
{
    cin >> a[i] >> b[i];
}
for(int i = 0; i < t; i++)
{
    cout << a[i] << b[i];
}
```



```
int t;
cin >> t;
while(t--)
{
    int a, b;
    cin >> a >> b;
    cout << a + b << '\n';
}
```

테스트케이스 하나 받고 정답 하나 출력하기!  
오른쪽 방법이 국룰입니다.



연습 문제  
10950번  $A+B - 3$

<https://www.acmicpc.net/problem/10950>

### 3. 입력의 개수를 모르는 경우

파일의 끝, EOF까지 반복하면 됩니다.  
그렇다면 EOF는 어떻게 판정할까요?

다음 슬라이드에서 세가지 방법을 알아보시다.

#### 입력

입력은 여러 개의 테스트 케이스로 이루어져 있다.

각 테스트 케이스는 한 줄로 이루어져 있으며, 각 줄에 A와 B가 주어진다. ( $0 < A, B < 10$ )

#### 출력

각 테스트 케이스마다 A+B를 출력한다.

#### 예제 입력 1 복사

```
1 1
2 3
3 4
9 8
5 2
```

#### 예제 출력 1 복사

```
2
5
7
17
7
```

### 3. 입력의 개수를 모르는 경우

```
while(cin << a << b)
{
    ...
}

while(scanf("%d %d", &a, &b) != EOF)
{
    ...
}

while(scanf("%d %d", &a &b) == 2) //scanf는 입력 받은 갯수를 반환합니다.
{
    ...
}
```

자신에게 맞는 방법을 이용하면 됩니다.  
입력 객체 혹은 함수의  
반환 값을 알아두면 유용합니다!





## 연습 문제

10951번 A+B - 4

<https://www.acmicpc.net/problem/10951>

## 힌트

본 문제는 EOF를 잘 처리할 수 있는지 확인하는 문제입니다.  
언어별로 EOF 레퍼런스에 차이가 있습니다. 한번 직접 찾아보세요!  
윈도우 환경에서는 터미널에서 Ctrl+Z를 눌러주면 EOF 처리가 됩니다.

## 4. 입력을 한 줄 씩 받아야 하는 경우

문장 안에 공백이 포함될 수 있습니다.  
String 자료형과 getline 함수를 이용하면  
편리하게 해결할 수 있습니다.

### 입력

입력이 주어진다. 입력은 최대 100줄로 이루어져 있고, 알파벳 소문자, 대문자, 공백, 숫자로만 이루어져 있다. 각 줄은 100글자를 넘지 않으며, 빈 줄은 주어지지 않는다. 또, 각 줄은 공백으로 시작하지 않고, 공백으로 끝나지 않는다.

### 출력

입력받은 그대로 출력한다.

#### 예제 입력 1 복사

```
Hello
Baekjoon
Online Judge
```

#### 예제 출력 1 복사

```
Hello
Baekjoon
Online Judge
```


## 4. 입력을 한 줄 씩 받아야 하는 경우

문장 안에 공백이 포함될 수 있습니다.  
String 자료형과 getline 함수를 이용하면  
편리하게 해결할 수 있습니다.

```
string str;
while (1)
{
    getline(cin, str);
    if (str=="")
        break;
    cout << str << endl;
}
```

## 4. 입력을 한 줄 씩 받아야 하는 경우

정규표현식을 이용한 방법도 있습니다.



```
while (scanf("%[^\n]\n",s)==1) {  
    printf("%s\n",s);  
}
```



연습 문제  
11718번 그대로 출력하기

<https://www.acmicpc.net/problem/11718>

입출력

**TIP!**

# 입출력 속도

입출력 방식에 따라 프로그램 실행 속도에 차이가 난다는 사실 알고 계셨나요?

printf/scanf가 cin/cout보다 월등히 빠릅니다.

입력 혹은 출력 데이터가 10만개 이상일시 적절한 입출력 방식을 생각해주어야 합니다.  
평소에 printf/scanf를 사용하는 습관을 들이는 것도 좋은 방법입니다.

printf/scanf > cin/cout

동작속도

# 입출력 속도

개행 방법에도 실행 속도에 차이가 있습니다.

‘\n’이 endl 객체보다 빠릅니다.

왜냐하면 endl 객체는 개행만 해주는 것이 아니라  
내부의 버퍼를 비워주는 역할도 하기 때문에 느립니다.  
사소하지만 알아두시다!

‘\n’ > endl

동작속도



읽을거리



<http://www.cplusplus.com/reference/cstdio/printf/>

<http://www.cplusplus.com/reference/cstdio/scanf/>

## printf/scanf 사용법 레퍼런스

”레퍼런스 문서를 애용합니다!  
겉보기에는 딱딱해 보여도  
도움이 되는 모든 정보를 담고 있습니다.”

꿀팁



전 그래도 cin/cout 사용하고 싶어요!



```
ios_base :: sync_with_stdio(false);  
cin.tie(NULL);  
cout.tie(NULL);
```

다음 코드를 추가해주시면 cin/cout을 scanf/printf의 속도로  
가속하여 사용할 수 있습니다.

단, 이 방법을 사용하면 cin/cout만을 사용하여야 하고,  
싱글 쓰레드 환경에서만 작동하기 때문에  
실제 개발에서는 사용하지 말아주세요!

꿀팁



입력 데이터를 일일이 쳐보기 귀찮아요!

<https://ideone.com/ideone/Index/submit/>

ideone 온라인 컴파일러 사이트에서  
입력 데이터를 한번에 붙여 넣고  
컴파일 할 수 있습니다!  
애용해주세요!

중요



분명 내 코드도 맞고 예제 출력도 맞는데  
틀렸다고 합니다!

**“코드는 거짓말하지 않습니다.”**

차근차근 틀린 이유를 살펴봅시다!

<https://www.acmicpc.net/blog/view/70>

백준 자주 틀리는 요인 정리

# The End

수고하셨습니다!

다음시간에는  
기초적인 자료구조인  
스택, 큐, 덱을 배워봅시다.