

---

# 실습문제의 해결 프로그램 작성 방법

**2023. 03**

국민대학교

소프트웨어융합대학



# 과제물 문제

---

- 실습문제 형식

구분	내용
문제이름	프로그래밍해야 할 실습문제의 이름
문제설명	프로그래밍해야 할 실습문제의 내용을 설명
예제	위에서 설명한 문제에 해당하는 간단한 예
입력 데이터	작성된 프로그래밍의 오류를 검증할 테스트 데이터의 입력 형식을 설명
출력 데이터	입력된 데이터의 해답을 출력하는 형식 설명
입력과 출력의 예	입력 데이터와 출력 데이터의 예를 제시
학습목표	이 문제를 통하여 배우게 되는 주된 학습 내용
주의사항	문제해결 시 주의해야 할 사항

# 과제물 문제의 예 (1/4)

문제이름	주어진 정수의 합 구하기
문제설명	주어진 정수들의 합을 계산하는 프로그램을 작성하시오.
예제	<p>예를 들어, 다음과 같은 세 개의 정수  <math>3 \ -4 \ 5</math>  의 합은, <math>3 + (-4) + 5 = 4</math> 이다. 또한, 다음과 같은 다섯 개  의 정수  <math>-8 \ -4 \ -6 \ -4 \ -10</math>  의 합은 <math>(-8) + (-4) + (-6) + (-4) + (-10) = -32</math> 이며, 그리고,  한 개의 정수  <math>10</math>  의 합은 10이다.</p>

# 과제물 문제의 예 (2/4)

<b>입력데이터</b>	<p><b>입력</b></p> <p>입력은 표준입력(standard input)을 사용한다. 입력은 <math>t</math> 개의 테스트 케이스로 주어진다. 입력의 첫 줄에 테스트 케이스의 개수를 나타내는 정수 <math>t</math>가 주어진다. 두 번째 줄부터 하나의 테스트 데이터에 대하여 두 줄씩 데이터가 입력된다. 각 테스트 케이스에 해당하는 첫 번째 줄에는 먼저 합을 계산하여야 할 정수의 개수를 나타내는 정수 <math>n</math> (<math>1 \leq n \leq 10,000</math>) 이 주어진다. 두 번째 줄에는 합을 계산하여야 할 <math>n</math> 개의 정수들이 한 줄에 주어진다. 이 정수들의 절대값은 100 보다 작거나 같다. 모든 정수들 사이에는 하나의 스페이스가 있다.</p>
--------------	---

# 과제물 문제의 예 (3/4)

---

출력데이터	<p>출력</p> <p>출력은 표준출력(standard output)을 사용한다. 입력 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는 모든 정수들의 합을 나타내는 정수를 출력한다.</p>
-------	---

# 과제물 문제의 예 (4/4)

<b>입력과 출력의 예</b>	<b>입력과 출력의 예</b> <table><tr><th data-bbox="537 471 1122 546">입력</th><th data-bbox="1122 471 1705 546">출력</th></tr><tr><td data-bbox="537 546 1122 611">3</td><td data-bbox="1122 546 1705 611">4</td></tr><tr><td data-bbox="537 611 1122 675">3</td><td data-bbox="1122 611 1705 675">-32</td></tr><tr><td data-bbox="537 675 1122 739">3 -4 5</td><td data-bbox="1122 675 1705 739">10</td></tr><tr><td data-bbox="537 739 1122 803">5</td><td data-bbox="1122 739 1705 803"></td></tr><tr><td data-bbox="537 803 1122 868">-8 -4 -6 -4 -10</td><td data-bbox="1122 803 1705 868"></td></tr><tr><td data-bbox="537 868 1122 932">1</td><td data-bbox="1122 868 1705 932"></td></tr><tr><td data-bbox="537 932 1122 996">10</td><td data-bbox="1122 932 1705 996"></td></tr></table>	입력	출력	3	4	3	-32	3 -4 5	10	5		-8 -4 -6 -4 -10		1		10	
입력	출력																
3	4																
3	-32																
3 -4 5	10																
5																	
-8 -4 -6 -4 -10																	
1																	
10																	
<b>학습목표</b>	정수 연산자에 대하여 학습한다.																
<b>주의사항</b>																	

# 입력 및 입력의 예 (1/3)

---

- 입력 부분에서는 문제를 해결하는 프로그램을 테스트하기 위한 입력 데이터에 관하여 설명한 부분이다.
  - a. 먼저 모든 입력 데이터는 표준입력(standard input)으로부터 입력된다. 표준입력은 key board를 의미한다.
    - 큰 입력 데이터를 테스트하기 위해서는 input.txt와 같은 텍스트 파일에 입력 데이터를 저장한 후에 파일 redirection을 이용해서 표준입력 대신 파일로부터 입력하도록 하면 된다.
  - b. 위의 문제 설명에서 세 가지의 예를 들어서 설명한 것과 같이, 입력 파일에는 여러 개의 테스트 데이터가 저장되어 있다. 입력 부분에서는 이러한 입력 데이터가 어떤 형태로 저장되어 있는지를 설명한다.
  - c. 입력파일의 첫 줄은 항상 입력 테스트 데이터의 개수를 나타내는 정수가 저장되어 있다.
  - d. 또한, 입력 부분에서는 입력의 개수의 범위 혹은 입력 데이터의 크기에 대한 제한 조건도 설명되어 있다.

## 입력 및 입력의 예 (2/3)

입력은 표준입력(standard input)을 사용한다.

테스트 데이터는 Key Board  
를 통해서 입력된다.  
(redirection할 수 있음)

입력은  $t$  개의 테스트 케이스로 주어진다. 입력의 첫 줄에 테스트 케이스의 개수를 나타내는 정수  $t$  가 주어진다.

입력파일의 첫 째 줄에  
입력 테스트 케이스의  
개수가 주어짐.

두 번째 줄부터 하나의 테스트 데이터에 대하여 두 줄씩  
데이터가 입력된다.

실제 테스트 데이터는  
둘째 줄부터 시작되며  
각 테스트 데이터는 두  
줄씩 주어짐.

각 테스트 케이스에 해당하는 첫 번째 줄에는 먼저 합을 계산  
하여야 할 정수의 개수를 나타내는 정수  $n$  ( $1 \leq n \leq 10,000$ ) 이  
주어진다.

각 테스트 데이터의 첫  
째 줄에는 그 테스트  
데이터에서 사용되는  
정수의 개수가 주어진다. 또한 정수의 개수  
의 범위가 설명되어 있  
다.

두 번째 줄에는 합을 계산하여야 할  $n$  개의 정수들이 한 줄에  
주어진다. 이 정수들의 절대값은 100 보다 작거나 같다. 모든  
정수들 사이에는 하나의 공백이 있다.

각 테스트 데이터의 두  
번째 줄에는 실제 데이  
터인 정수가 주어지며,  
이 정수들의 범위가 주  
어진다.



# 입력 및 입력의 예 (3/3)

입력은 표준입력(standard input)을 사용한다.

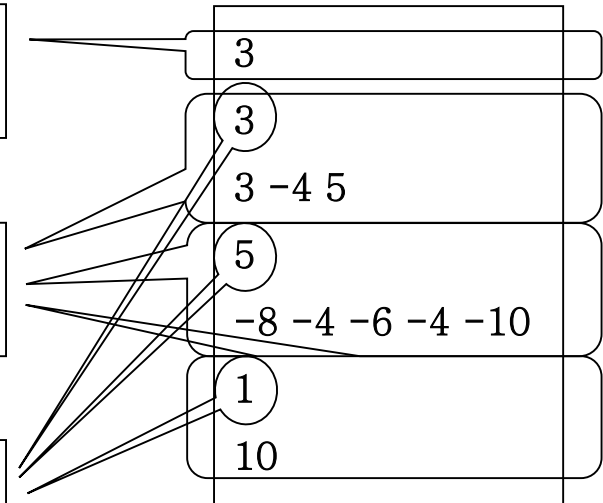
입력은  $t$  개의 테스트 케이스로 주어진다. 입력의 첫 줄에 테스트 케이스의 개수를 나타내는 정수  $t$  가 주어진다.

두 번째 줄부터 하나의 테스트 데이터에 대하여 두 줄씩 데이터가 입력된다.

각 테스트 케이스에 해당하는 첫 번째 줄에는 먼저 합을 계산하여야 할 정수의 개수를 나타내는 정수  $n$  ( $1 \leq n \leq 10,000$ ) 이 주어진다.

두 번째 줄에는 합을 계산하여야 할  $n$  개의 정수들이 한 줄에 주어진다. 이 정수들의 절대값은 100 보다 작거나 같다. 모든 정수들 사이에는 하나의 공백이 있다.

## 입력의 예



# 출력 및 출력의 예 (1/2)

---

- 출력 부분에서는 문제를 해결하는 프로그램에서 각 입력 데이터에 대한 해답을 출력하는 형식에 관한 설명을 하는 부분이다.
  - a. 먼저 모든 출력 데이터는 표준출력(standard output)을 사용한다.
  - b. 각 입력 테스트 데이터에 대한 해답을 출력하는 형식에 대하여 설명한다.

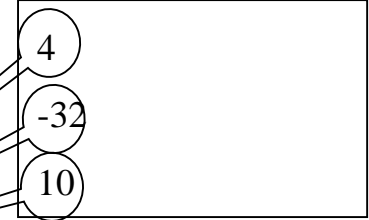
## 출력 및 출력의 예 (2/2)

출력은 표준출력(standard output)을 사용한다.

입력 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트의 결과를 출력한다.

각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는 모든 정수들의 합을 나타내는 정수를 출력한다.

출력의 예



# 프로그램 작성의 예 (1/2)

```
/* *****  
*  
* Problem: 주어진 정수의 합 구하기  
*  
* ***** */  
  
/* *****  
*  
* 국민대학교 소프트웨어융합대학 소프트웨어학부 2학년  
*  
* 20217777 홍길동  
*  
* ***** */
```

프로그램에 대한 간단한 설명과 프로그램작성자에 대한 주석은 프로그램의 최상단부에 둔다.

```
#include <iostream>  
using namespace std;
```

함수 main()은 int 자료형을 리턴하도록 정의한다.

```
int main(void)  
{  
    int numTestCases;
```

입력파일의 첫 번째 줄에서 입력 테스트 케이스의 개수를 입력한다.

```
/* read the number of test cases */  
cin >> numTestCases;
```

## 프로그램 작성의 예 (2/2)

```
for(int i=0; i<numTestCases; i++)  
{
```

각 테스트 케이스를 반복적으로 처리한다.

```
    int numData, data;  
    int sum = 0;
```

```
    cin >> numData;
```

각 테스트 케이스의 첫 번째 줄에 있는  
입력되는 정수의 개수를 입력한다.

```
    for (int j=0; j<numData; j++)  
    {
```

각 테스트 케이스의 모든 정수를 입력하고 그 정  
수들의 합을 계산한다.

```
        cin >> data;  
        sum += data;  
    }
```

```
    cout << sum << endl; /* print out the sum of numbers */
```

```
}  
  
return 0;
```

각 테스트 케이스의 해답을 출력한다.

main()는 항상 0을 return 하도록 한다.

```
3  
3  
3 -4 5  
5  
-8 -4 -6 -4 -10  
1  
10
```

# 프로그램 작성시 주의할 점

---

- 함수 `main()` 을 정의할 때
  - 다음의 예와 같이 `int` 자료형을 리턴하도록 정의한다.
  - 마지막에는 `0`을 리턴하도록 작성한다.

```
int main()
{
    return 0;
}
```

# 프로그램 작성시 주의할 점

---

- 프로그램에서 데이터를 입력할 때, 다음 예와 같이 설명을 해 줄 필요가 없다.

```
cout << “데이터를 입력하시오.\n”;
```

- 또한 해답을 출력할 경우에도 다음 예와 같은 문장을 출력할 필요가 없다.

```
cout << “해답은 다음과 같습니다.\n”;
```

# 해답 출력 시 주의할 점

---

- 해답을 출력할 때

- 해답은 각 줄의 첫 번째 열부터 출력한다. 즉 각 줄의 첫 번째 열에 공백을 두어서는 안된다.
- 한 줄에 여러 개의 해답 데이터(정수, 문자, 문자열 등)를 출력하는 경우에, 각 데이터 사이에는 하나의 스페이스만을 두어야 한다.
- 단, 각 줄에서 데이터를 출력한 다음에는 공백 문자를 출력하여도 된다.



# 주의할 점

---

- 프로그램 정확도 테스트

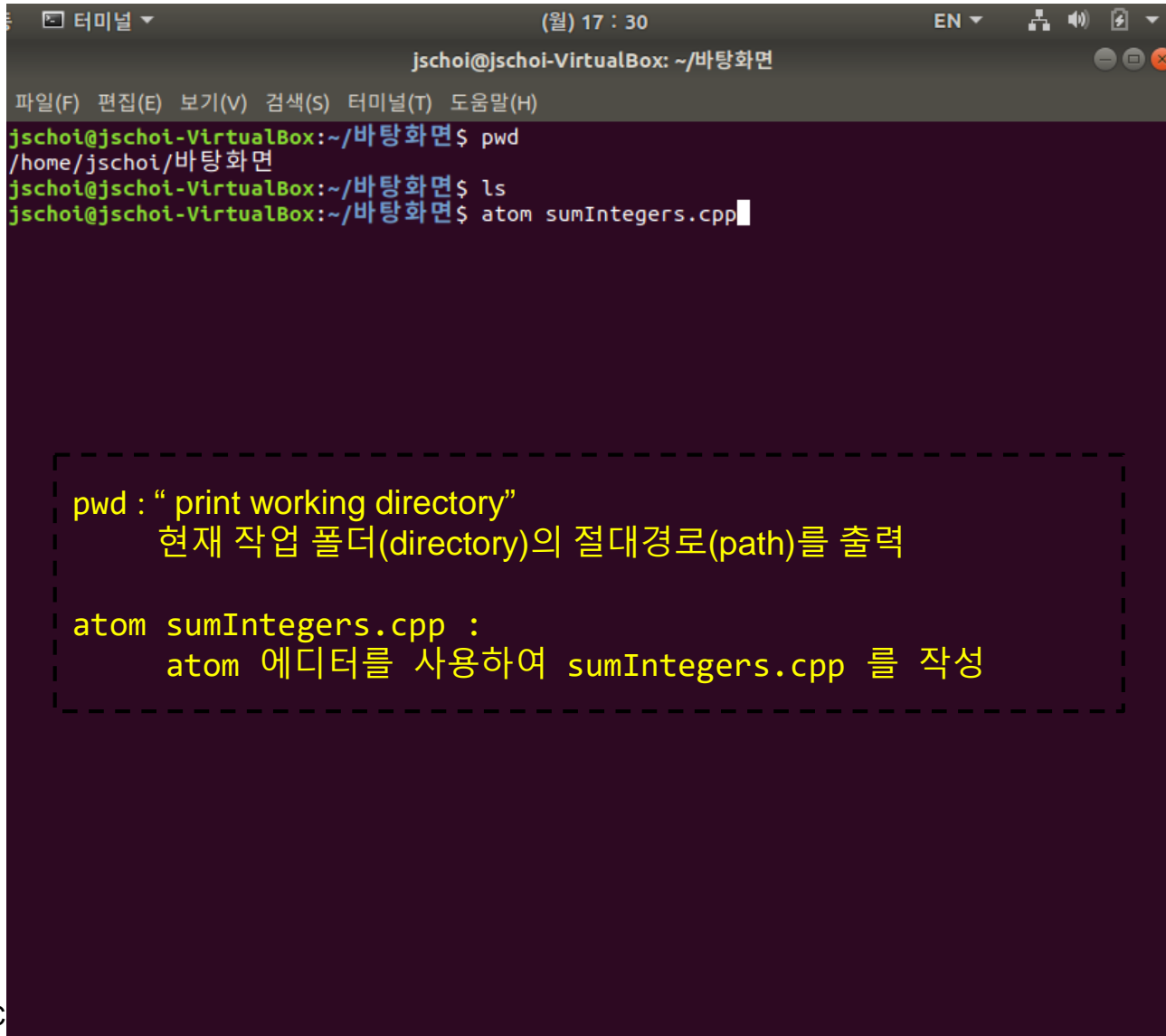
- 문제에서 제시된 입력과 출력의 예는 **단지 예시일 뿐**이다.
- 예시된 테스트 데이터에 대해서 모두 정답을 출력하였다고 해서 프로그램이 정확하다는 의미는 아니다.
- 채점 서버에는 좀 더 복잡한 테스트 데이터를 사용하여 제출된 프로그램을 테스트한다.

---

# Ubuntu 에서 실습



# Ubuntu 에서 실습



The screenshot shows a terminal window titled 'jschoi@jschoi-VirtualBox: ~/바탕화면'. The terminal displays the following commands and their outputs:

```
jschoi@jschoi-VirtualBox:~/바탕화면$ pwd
/home/jschoi/바탕화면
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
jschoi@jschoi-VirtualBox:~/바탕화면$ atom sumIntegers.cpp
```

Below the terminal output, a dashed box contains the following explanations:

- `pwd` : “ print working directory”  
현재 작업 폴더(directory)의 절대경로(path)를 출력
- `atom sumIntegers.cpp` :  
atom 에디터를 사용하여 `sumIntegers.cpp` 를 작성

# Ubuntu 에서 실습

```
1  /*
2   * Sum of Integers
3   */
4
5  #include <iostream>
6  using namespace std;
7
8  int main()
9  {
10     int numTestCases;
11
12     cin >> numTestCases;
13     for(int i=0; i<numTestCases; i++)
14     {
15         int numData, data;
16         int sum = 0;
17
18         cin >> numData;
19         for(int j=0; j<numData; j++)
20         {
21             cin >> data;
22             sum += data;
23         }
24         cout << sum << endl;
25     }
26     return 0;
27 }
```

# Ubuntu 에서 실습

- 표준입력(keyboard)에서 직접 입력데이터 입력하는 예

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$
```

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ sumIntegers
sumIntegers: 명령을 찾을 수 없습니다
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers
```

# Ubuntu 에서 실습

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ sumIntegers
sumIntegers: 명령을 찾을 수 없습니다
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers
3
3
3 -4 5
4
```

입력의 첫 번째 줄 (입력 데이터의 개수)

첫 번째 입력데이터

첫 번째 입력데이터의 출력

# Ubuntu 에서 실습

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ sumIntegers
sumIntegers: 명령을 찾을 수 없습니다
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers
3
3
3 -4 5
4
5
-8 -4 -6 -4 -10
-32
```

두 번째 입력데이터

두 번째 입력데이터의 출력

# Ubuntu 에서 실습

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ g++ -o sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ sumIntegers
sumIntegers: 명령을 찾을 수 없습니다
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers
3
3
3 -4 5
4
5
-8 -4 -6 -4 -10
-32
1
10
10
jschoi@jschoi-VirtualBox:~/바탕화면$
```

세 번째 입력데이터

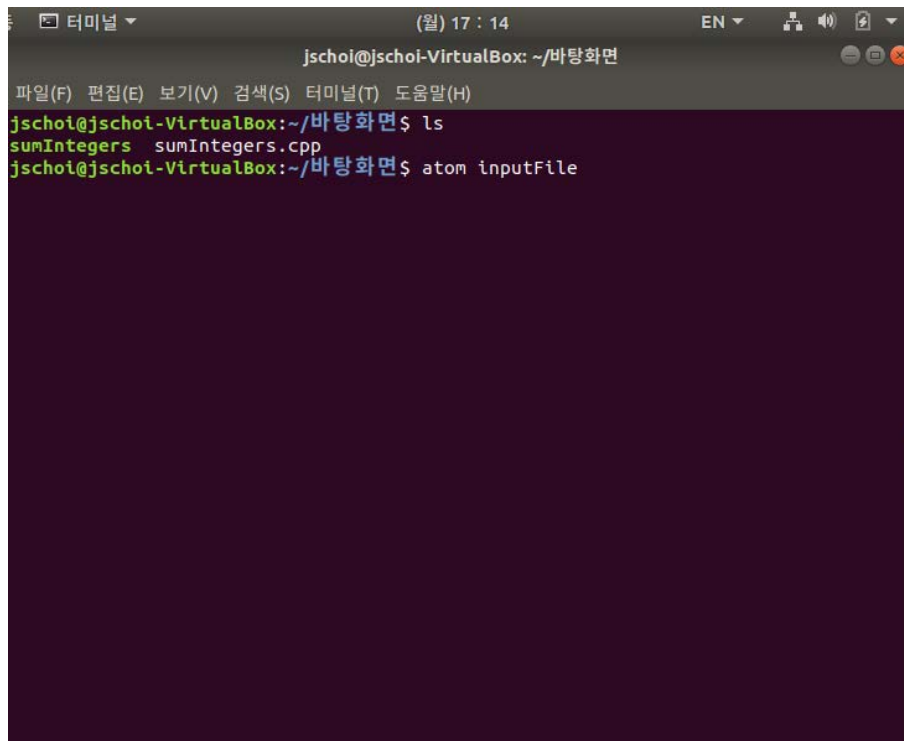
세 번째 입력데이터의 출력

-- 그리고 프로그램이 종료 됨 --



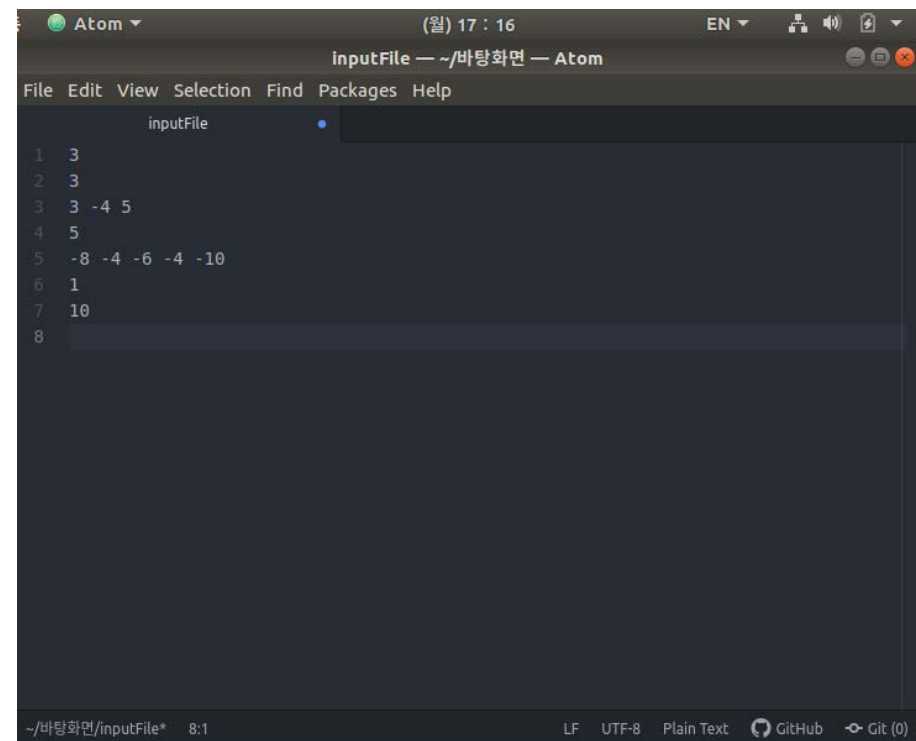
# Ubuntu 에서 실습

- 표준입력 재지정(input redirection)하여 파일에서 입력하는 경우
  - 먼저 입력데이터 파일(예: "inputFile")을 만듦



A terminal window titled 'jschoi@jschoi-VirtualBox: ~/바탕화면' with a timestamp of '(월) 17:14'. The window shows the following commands and output:

```
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers  sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ atom inputFile
```



An Atom editor window titled 'inputFile — ~/바탕화면 — Atom' with a timestamp of '(월) 17:16'. The window shows the content of the 'inputFile' file:

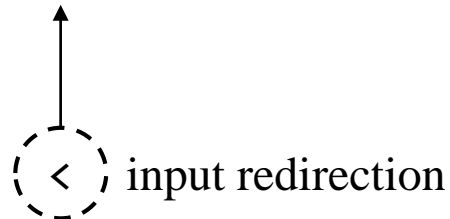
```
1 3
2 3
3 3 -4 5
4 5
5 -8 -4 -6 -4 -10
6 1
7 10
8
```

# Ubuntu 에서 실습

---

- 표준입력 재지정(input redirection)
  - 표준입력인 keyboard 대신에 파일(파일명:“inputFile”)에서 입력하고, 출력은 표준출력(모니터)로 출력함

```
sumIntegers < inputFile
```



# Ubuntu 에서 실습

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ atom inputFile
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
inputFile sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers < inputFile
4
-32
10
jschoi@jschoi-VirtualBox:~/바탕화면$
```

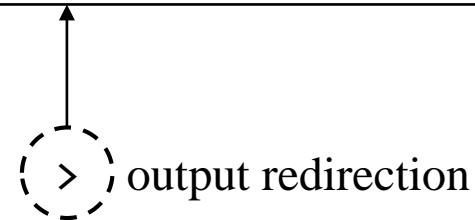
입력데이터가 “inputFile”로 부터 입력되며,  
출력은 모니터로 출력됨

# Ubuntu 에서 실습

---

- 표준입력/출력 재지정(input/output redirection)
  - 표준입력인 keyboard 대신에 파일(파일명:“inputFile”)에서 입력하고, 표준출력인 모니터 대신에 파일(파일명:“outputFile”)로 출력하도록 함

```
sumIntegers < inputFile > outputFile
```



# Ubuntu 에서 실습

```
jschoi@jschoi-VirtualBox: ~/바탕화면
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ atom input
jschoi@jschoi-VirtualBox:~/바탕화면$ ls
input sumIntegers sumIntegers.cpp
jschoi@jschoi-VirtualBox:~/바탕화면$ sumIntegers < input
sumIntegers: 명령을 찾을 수 없습니다
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers < input
4
-32
10
jschoi@jschoi-VirtualBox:~/바탕화면$ ./sumIntegers < input > output
jschoi@jschoi-VirtualBox:~/바탕화면$ cat output
4
-32
10
jschoi@jschoi-VirtualBox:~/바탕화면$
```

출력파일 "outputFile"의 내용을 모니터로 출력함

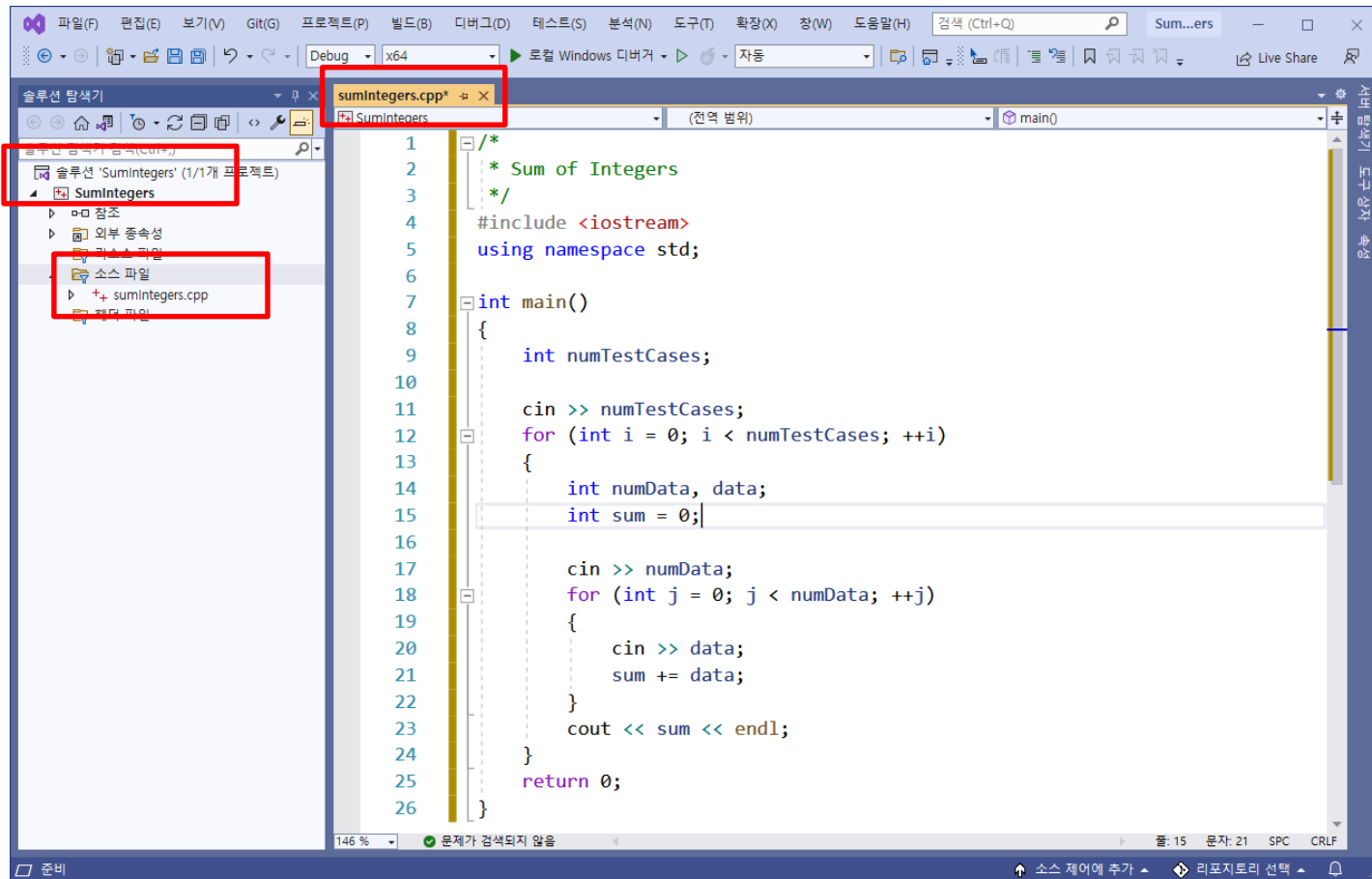
입력데이터가 "inputFile"로 부터 입력되며, 출력은 "outputFile"로 출력됨

# Microsoft Visual Studio 에서 실습



# MS Visual Studio에서 실습

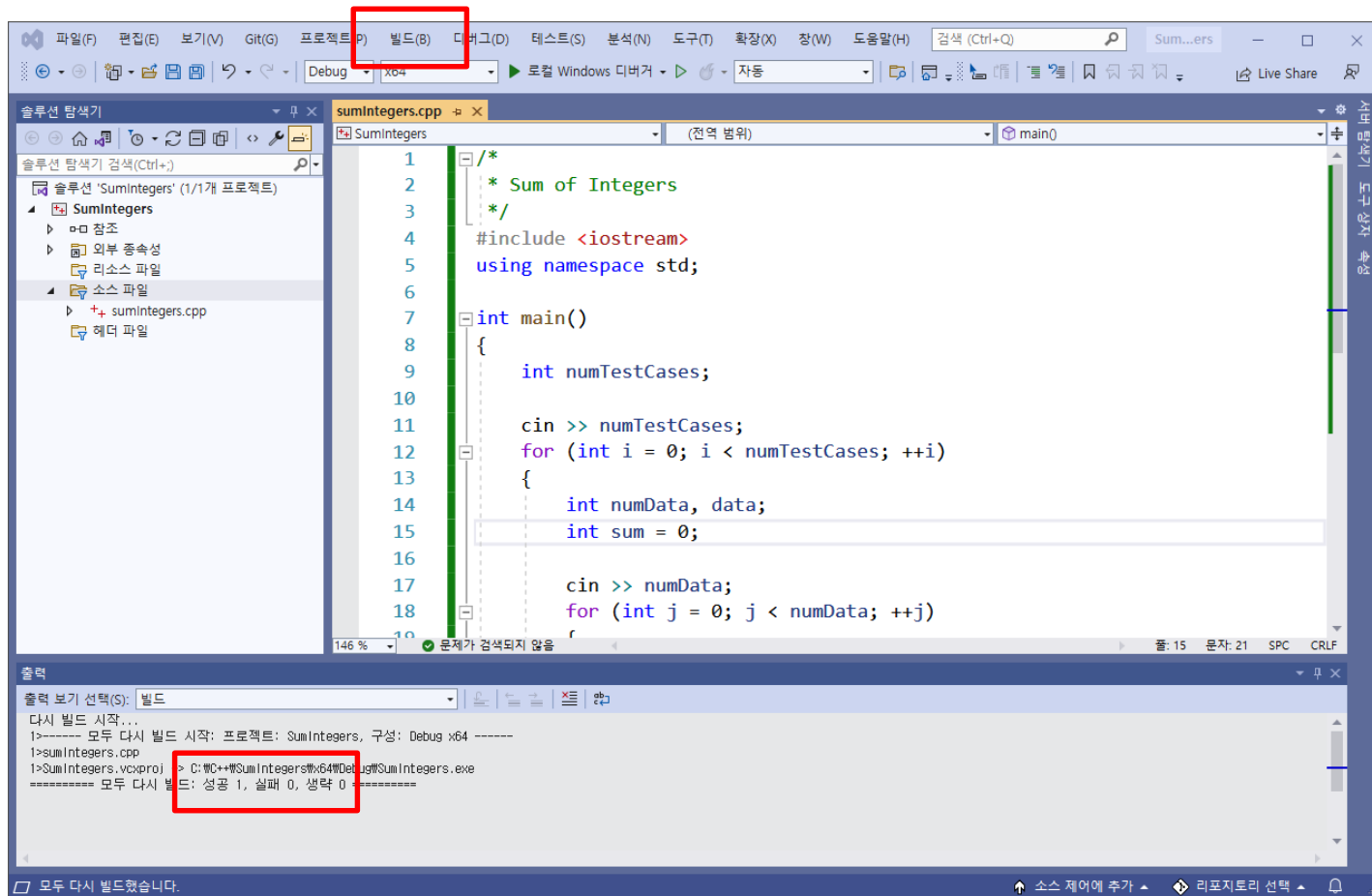
- 프로젝트에서 소스 파일 작성



# MS Visual Studio에서 실습

- 컴파일

- 메뉴: 빌드 -> 솔루션 빌드 (F7)

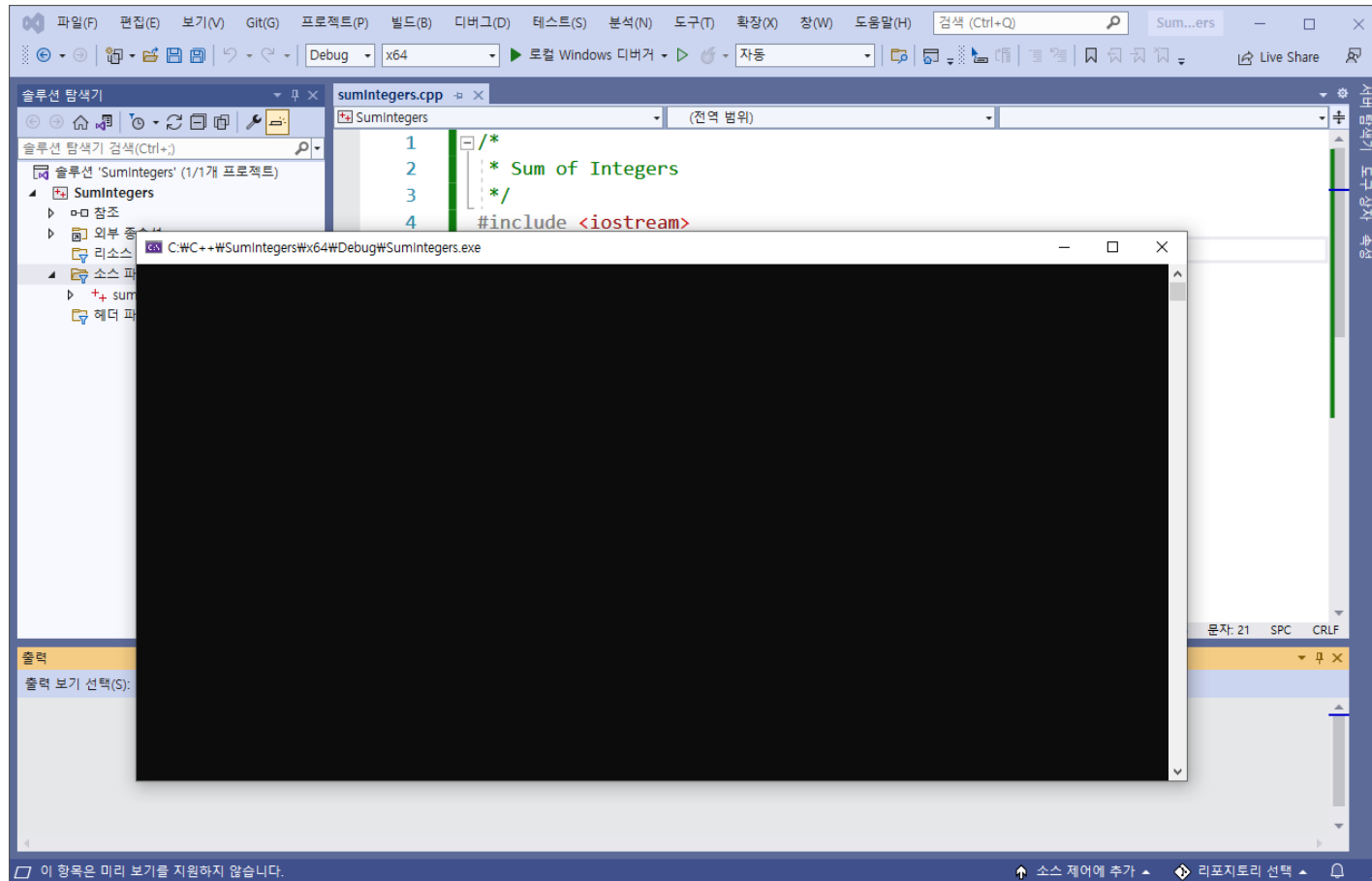




# MS Visual Studio에서 실습

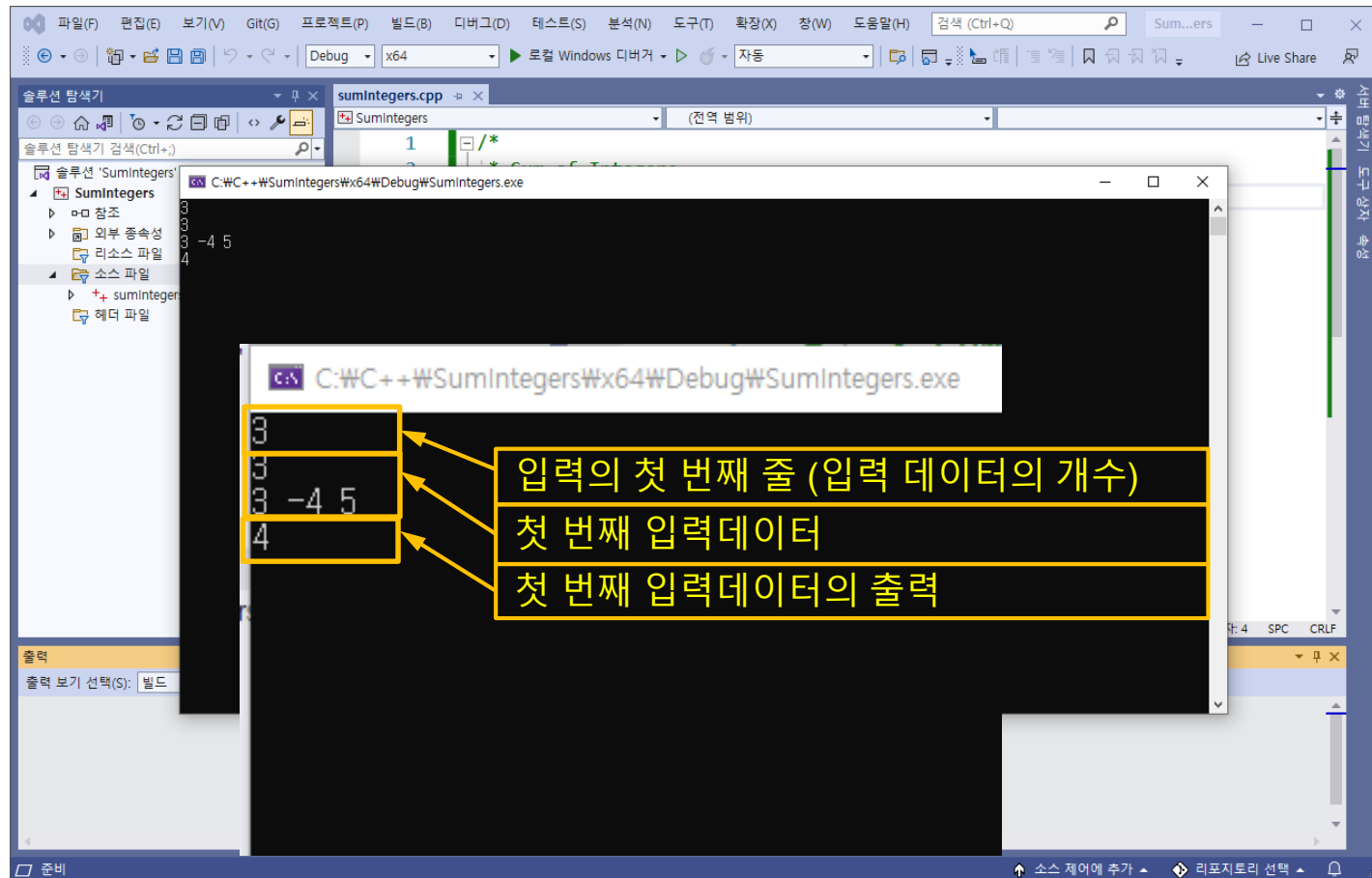
- 실행

- 메뉴: 디버그 -> 디버그 하지않고 시작 (Ctrl+F5)



# MS Visual Studio에서 실습

- 표준입력(keyboard)에서 직접 입력데이터 입력하는 예

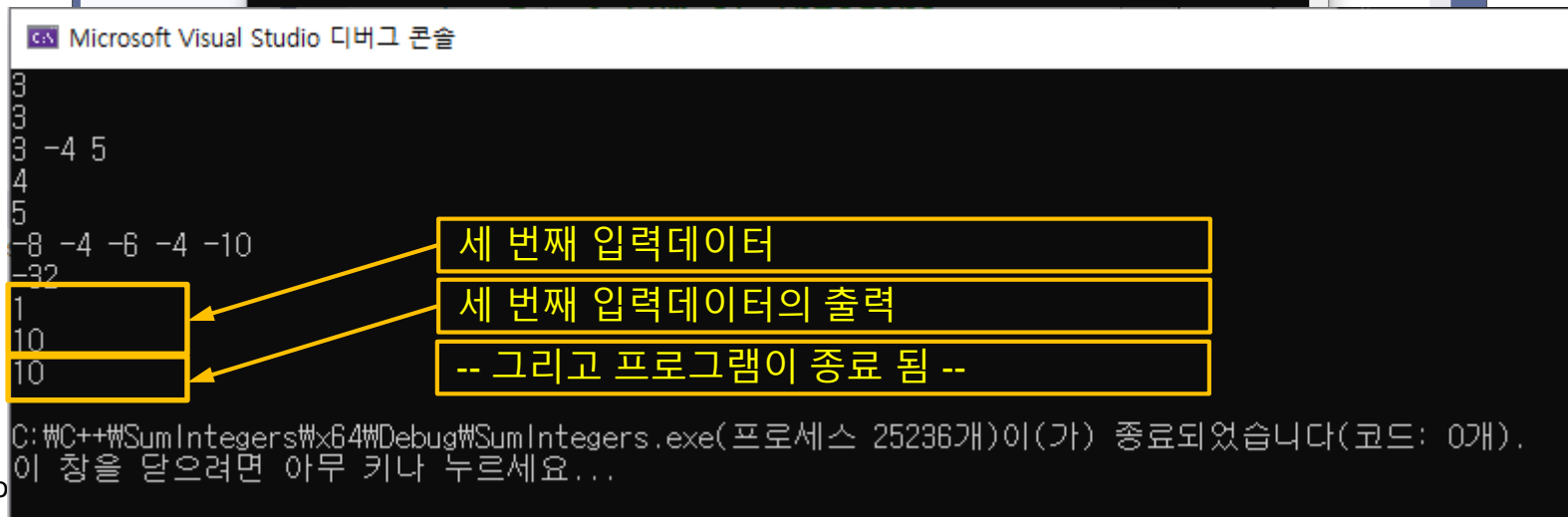
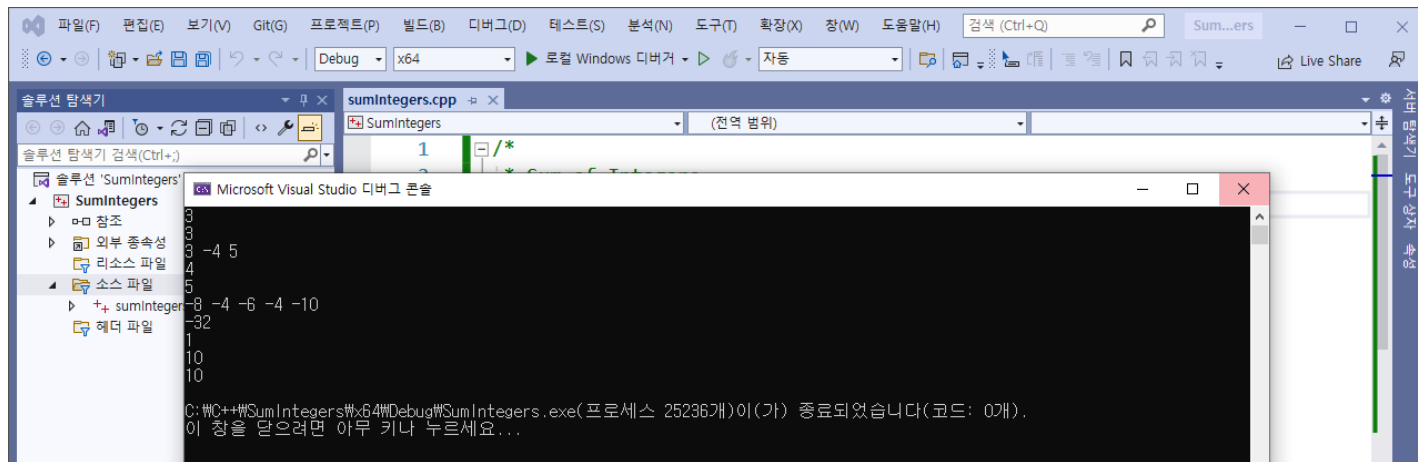


- 표준입력(keyboard)에서 직접 입력데이터 입력하는 예



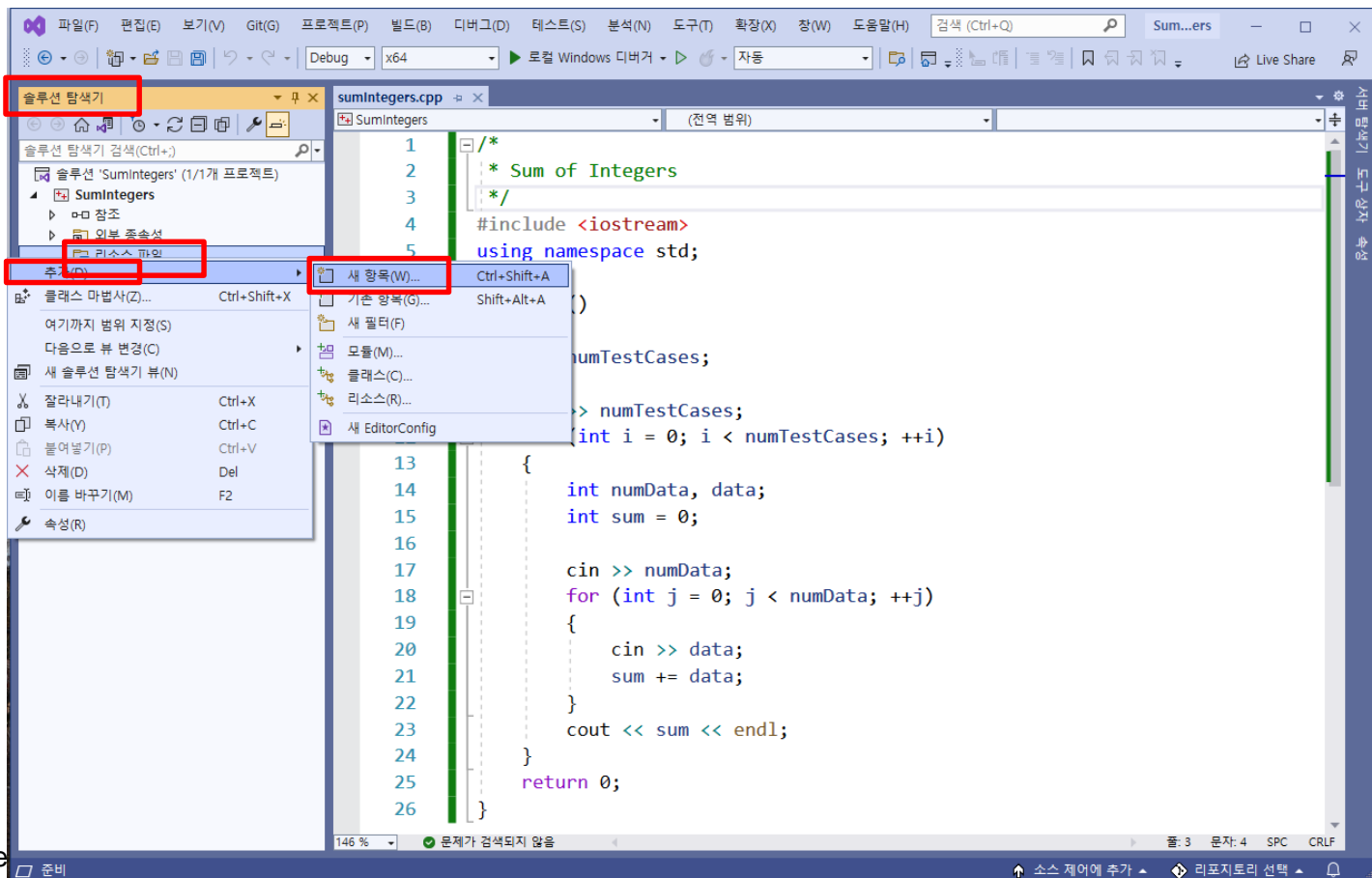
# MS Visual Studio에서 실습

- 표준입력(keyboard)에서 직접 입력데이터 입력하는 예



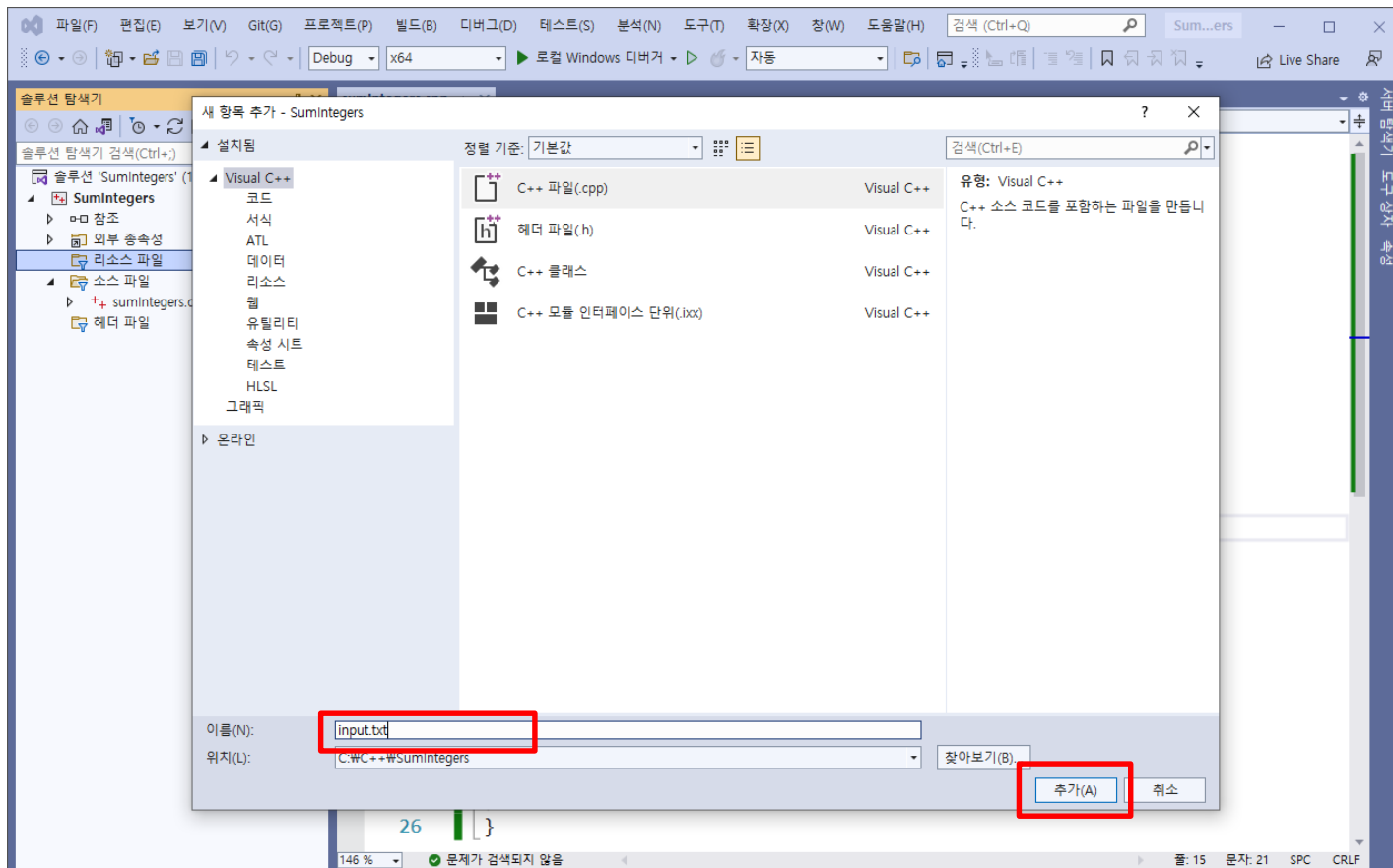
# MS Visual Studio에서 실습

- 표준입력 재지정(input redirection)하여 파일에서 입력하는 경우
  - 먼저 입력데이터 파일(예: "input.txt")을 만들 사용)
  - 메뉴 : 솔루션 탐색기 -> 리소스 파일 (마우스 우클릭) -> 추가 -> 새항목



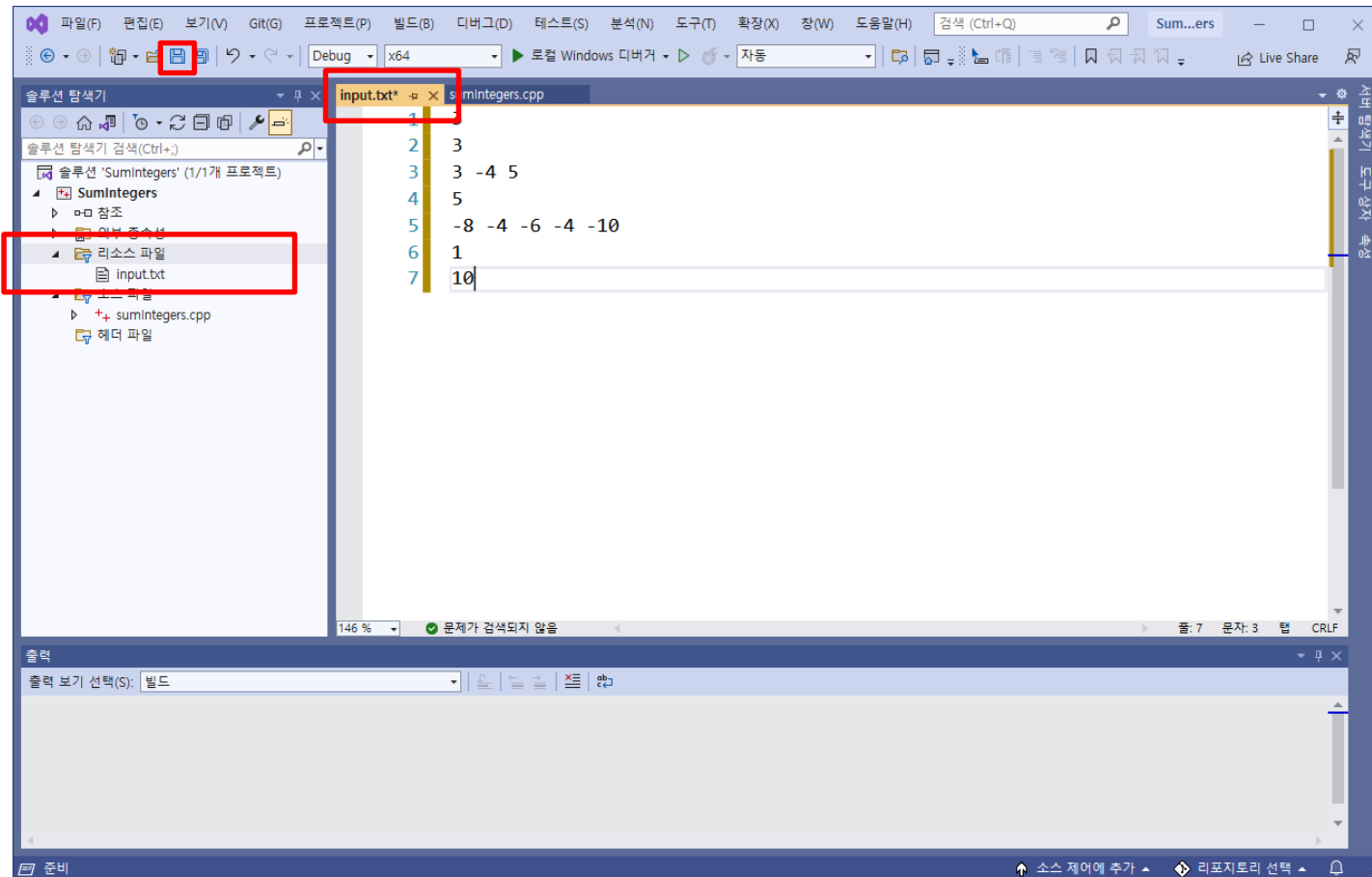
# MS Visual Studio에서 실습

- 표준입력 재지정(input redirection)하여 파일에서 입력하는 경우
  - 먼저 입력데이터 파일(예: "input.txt")을 만들 사용)
  - 메뉴 : 이름 항목에 "input.txt"를 입력함 -> 추가



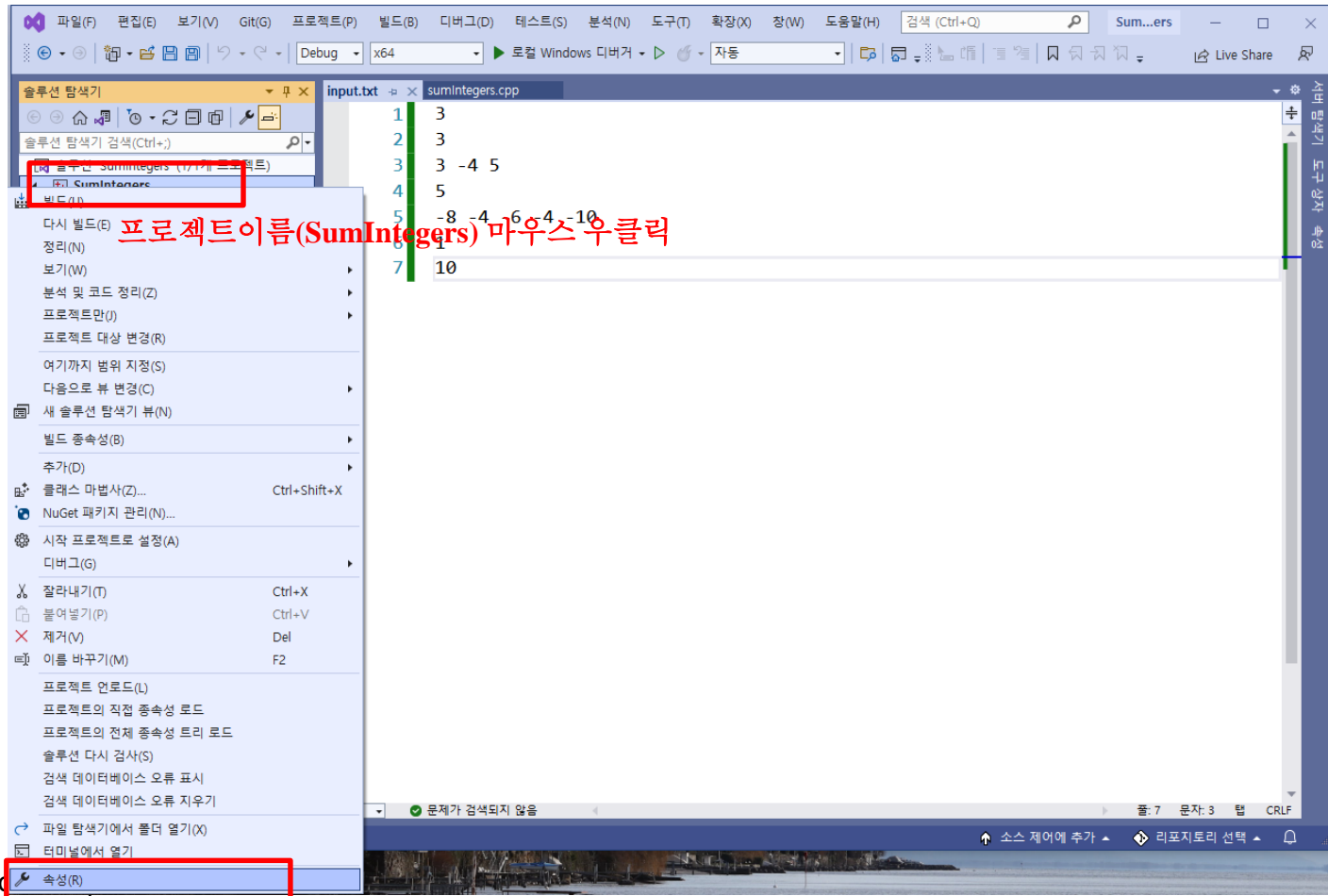
# MS Visual Studio에서 실습

- 표준입력 재지정(input redirection)하여 파일에서 입력하는 경우
  - 먼저 입력데이터 파일(예: “input.txt”)을 만들 사용) -> 파일저장



# MS Visual Studio에서 실습

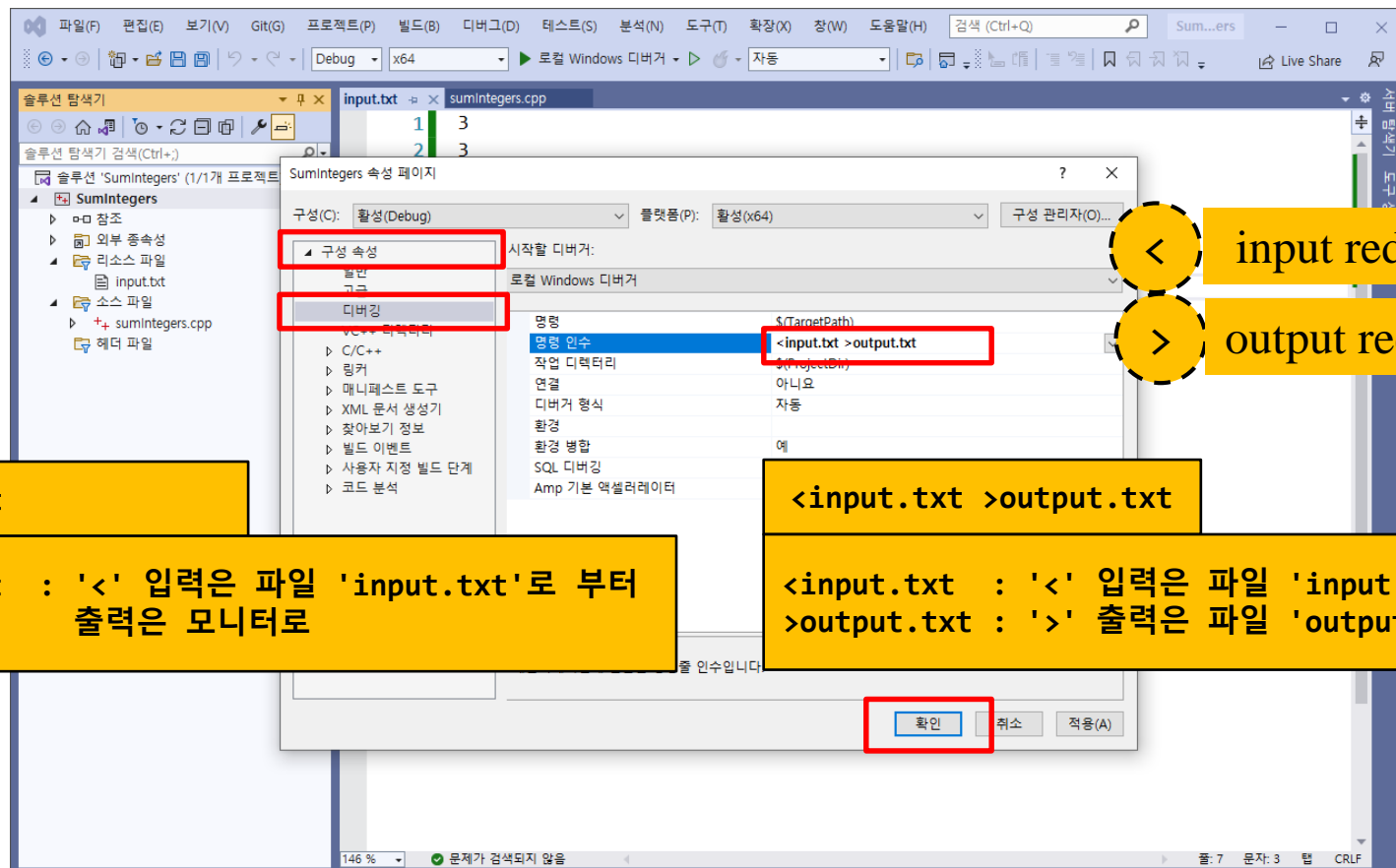
- 표준입력 재지정(input redirection) 설정
  - 메뉴: 프로젝트이름 (마우스 우클릭) -> 속성





# MS Visual Studio에서 실습

- 표준입력 재지정(input redirection) 설정
  - 메뉴: 프로젝트이름(마우스 우클릭) -> 속성 -> 구성 속성 -> 디버깅 -> 명령인수



# MS Visual Studio에서 실습

- 프로그램 실행후 출력파일 확인

