

| C언어 |

1주차

index

1. 프로젝트 준비

4. 자료형

2. 입출력

5. 연산자

3. 변수

6. 기타

1. 프로젝트 준비



프로젝트 준비

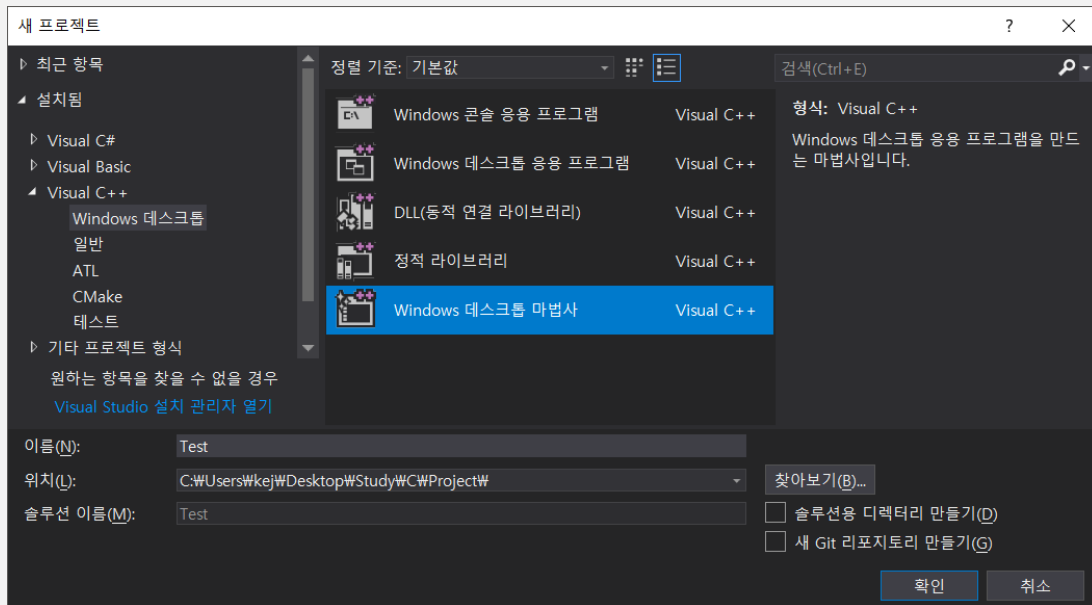
프로젝트 생성

소스 파일 생성

기본 코드 형식

| Visual Studio 2017

- Windows 데스크톱 마법사 선택
(혹은 Windows 콘솔 응용 프로그램 선택)





프로젝트 준비

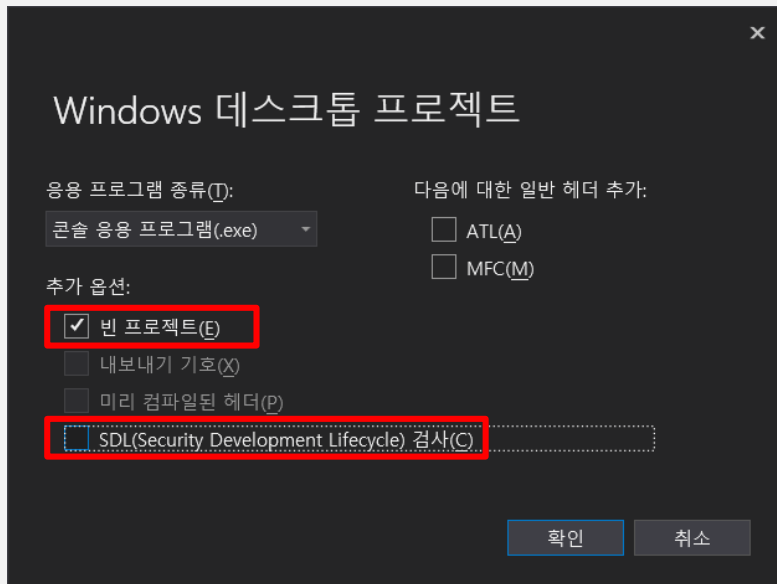
프로젝트 생성

소스 파일 생성

기본 코드 형식

| Visual Studio 2017

- 빈 프로젝트 선택
- SDL 검사 선택 해제





프로젝트 준비

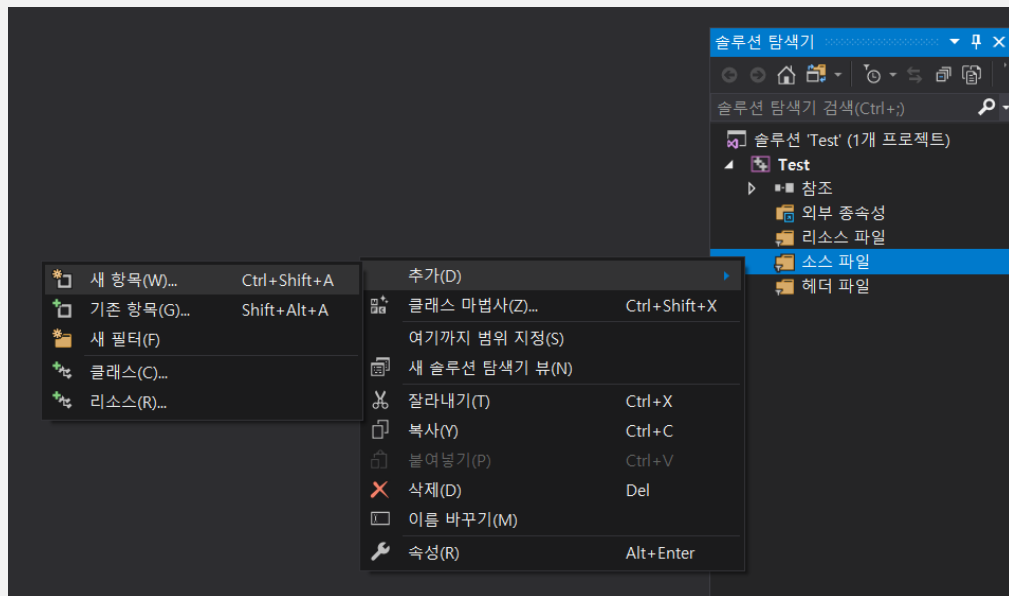
프로젝트 생성

소스 파일 생성

기본 코드 형식

| Visual Studio 2017

- 솔루션 탐색기 > 소스 파일 > 추가 > 새 항목





프로젝트 준비

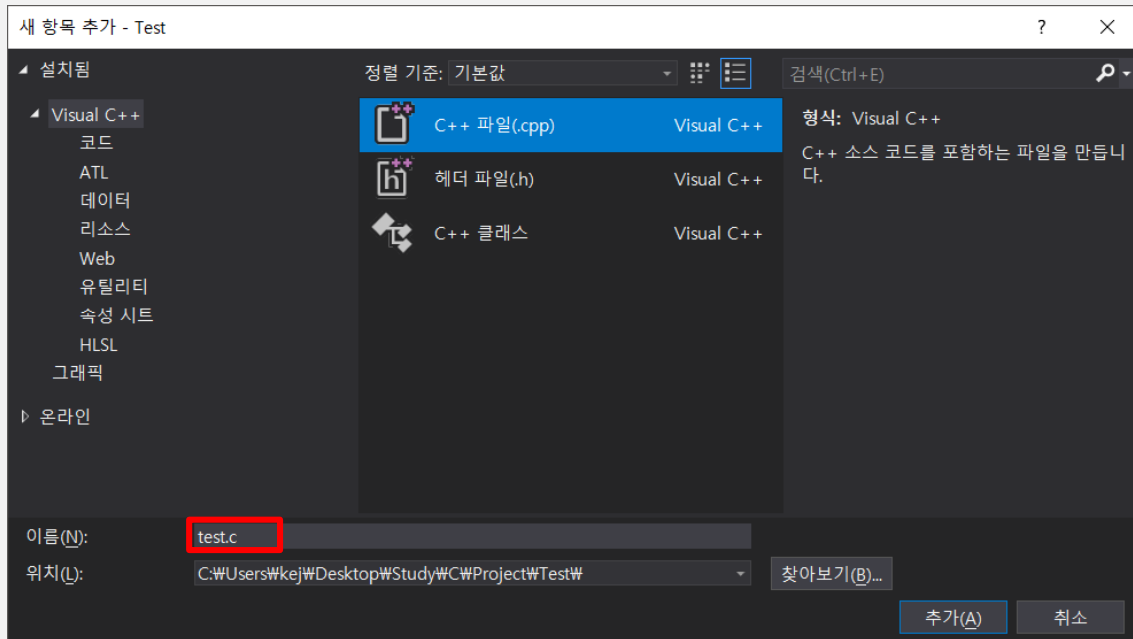
프로젝트 생성

소스 파일 생성

기본 코드 형식

| Visual Studio 2017

- C++ 파일(.cpp) 선택
- 파일 이름의 확장자를 .cpp에서 **.c**로 변경





프로젝트 준비

프로젝트 생성

소스 파일 생성

기본 코드 형식

| 기본 코드 형식

```
#include <stdio.h>

int main() {
    코드 내용

    return 0;
}
```

- 해당 코드의 ‘코드 내용’ 부분에 코드 작성
- 코드는 한 줄마다 끝에 ‘;’ 필요

2. 입출력



입출력

출력

입력

I printf

```
#include <stdio.h>

int main() {
    printf("Hello World!\n");

    return 0;
}
```



입출력

출력

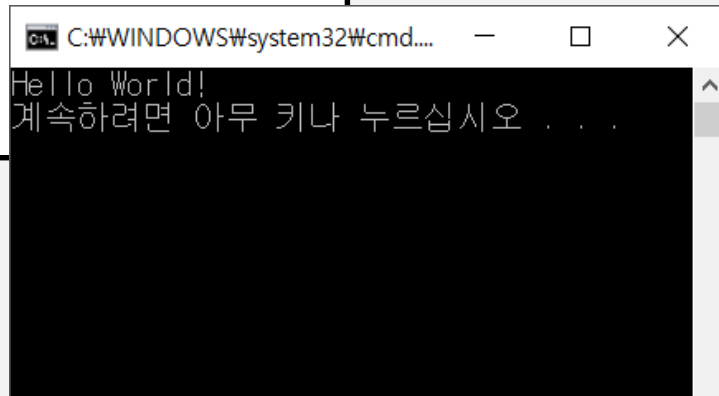
입력

I printf

```
#include <stdio.h>

int main() {
    printf("Hello World!\n");

    return 0;
}
```





입출력

출력

입력

| scanf

```
#include <stdio.h>
int main() {
    int a;

    scanf("%d", &a);

    printf("%d\n", a);

    return 0;
}
```



입출력

출력

입력

| scanf

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd...'. The window has a black background with white text. It displays the number '4' on two separate lines. Below the second '4', there is a line of Korean text: '계속하려면 아무 키나 누르십시오 . . .', followed by a cursor. The window includes standard Windows window controls (minimize, maximize, close) and a vertical scrollbar on the right side.

3. 변수



변수

선언 및 초기화
사용

| 선언 및 초기화

자료형 변수 이름 = 초기값;

자료형 변수 이름;

변수 이름 = 초기값;

- 2가지 방법 모두 가능
- 선언 후 초기화 혹은 대입 필요



변수

선언 및 초기화
사용

| 사용

- 초기화 없이 사용하면,

```
#include <stdio.h>
int main() {
    int a;

    printf("%d\n", a);

    return 0;
}
```

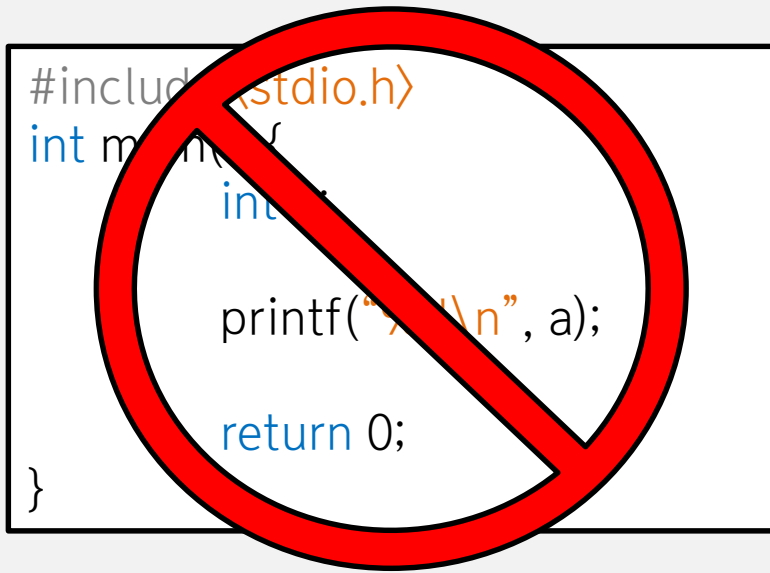



변수

선언 및 초기화
사용

| 사용

- 초기화 없이 사용하면, **쓰레기값 출력 및 빌드 오류 발생**



4. 자료형



자료형

종류

정수형

실수형

문자형

기타

| 자료형 종류

- 정수형: int
- 실수형: float, double
- 문자형: char
- 기타: short, long, signed, unsigned, ...

| 정수형 변수

- int: %d로 입력, %d로 출력
(4byte)

```
#include <stdio.h>
int main() {
    int a;

    scanf("%d", &a);

    printf("%d\n", a);

    return 0;
}
```



자료형

종류

정수형

실수형

문자형

기타

| 실수형 변수

- float: %f로 입력, %f로 출력
(4byte)

```
#include <stdio.h>
int main() {
    float a;

    scanf("%f", &a);

    printf("%f\n", a);

    return 0;
}
```



자료형

종류

정수형

실수형

문자형

기타

| 실수형 변수

- double: %lf로 입력, %lf 혹은 %f로 출력 (8byte)

```
#include <stdio.h>
int main() {
    double a;

    scanf("%lf", &a);

    printf("%lf\n", a);

    return 0;
}
```



자료형

종류
정수형
실수형
문자형
기타

| float와 double

- double의 정확도가 float보다 높음
- float는 소수점 아래 6자리,
double은 소수점 아래 15자리까지 표현 가능
- 상수로 실수를 사용할 때, f가 붙지 않으면
자동으로 double형으로 인식



자료형

종류

정수형

실수형

문자형

기타

| 문자형 변수

- char: %c로 입력, %c로 출력 (1byte)

```
#include <stdio.h>
int main() {
    char a;

    scanf("%c", &a);

    printf("%c\n", a);

    return 0;
}
```




자료형

종류

정수형

실수형

문자형

기타

| 기타 자료형

- short, long, long long, unsigned, signed, ...
- 기존 자료형(정수, 실수, 문자)과 함께 변수 선언 시 적용



자료형

종류

정수형

실수형

문자형

기타

| 부호 지정

- signed: 음수 ~ 양수
- unsigned: 0 ~ 양수

unsigned 사용하면 양수의 범위 2배로 증가
(음수 범위에서 사용하는 메모리를 양수 범위에 할당)



자료형

종류

정수형

실수형

문자형

기타

| 크기 지정

- short: 2byte
- long: 4byte
- long long: 8byte

기존 자료형에서 할당되는 기본 메모리 크기가 아닌,
해당 크기 지정 자료형의 메모리 크기로 할당

5. 연산자



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 연산자 종류

- 산술 연산자: +, -, *, /, %
- 할당 연산자: =, +=, -=, ...
- 증감 연산자: ++, --
- 관계 연산자: <, >, <=, >=, ==, !=
- 논리 연산자: &&, ||, !
- 조건 연산자: ?
- 비트 연산자: &, |, ~, ^, >>, <<



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 산술

- $A + B$: A와 B의 값을 더하기
- $A - B$: A에서 B의 값을 빼기
- $A * B$: A와 B의 값을 곱하기
- A / B : A에서 B의 값을 나누기
- $A \% B$: A / B 의 나머지

연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 산술

- 아래 코드를 실행하면 c의 값은?

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 10;
    int c = b % a;

    printf("c: %d\n", c);

    return 0;
}
```



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 할당

- $A = B$: A에 B의 값을 저장
- $A += B$: $A = A + B$ 와 같음
- $A -= B$: $A = A - B$ 와 같음
- $A *= B$: $A = A * B$ 와 같음
- $A /= B$: $A = A / B$ 와 같음
- $A \% = B$: $A = A \% B$ 와 같음

| 할당

- 아래 코드를 실행하면 a, b의 값은?

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 5;

    b = a;

    printf("a: %d\nb: %d\n", a, b);

    return 0;
}
```



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 할당

- 아래 코드를 실행하면 a, b의 값은?

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 4;

    a += b;

    printf("a: %d\nb: %d\n", a, b);

    return 0;
}
```



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 증감

- $A++$: 변수 A의 값을 사용한 후, A에 1 더하기
- $++A$: 변수 A에 1을 더한 후, A의 값을 사용
- $A--$: 변수 A의 값을 사용한 후, A에 1 빼기
- $--A$: 변수 A에 1을 뺀 후, A의 값을 사용

| 증감

- 아래 코드를 실행하면 출력되는 변수들의 값은?

```
#include <stdio.h>
int main() {
    int a = 4, b = 3;
    int c = 7, d = 10;

    printf("a: %d\nb: %d\n", a++, ++b);
    printf("a: %d\nb: %d\n", c--, --d);

    return 0;
}
```



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 관계

- 관계 연산의 결과가 틀리면 0, 맞으면 양수(보통 1)
- $A < B$: A가 B보다 작음
- $A > B$: A가 B보다 큼
- $A <= B$: A가 B보다 작거나 음
- $A >= B$: A가 B보다 크거나 같음
- $A == B$: A와 B의 값이 같음
- $A != B$: A와 B의 값이 틀림

연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 관계

- $b \geq a$ 의 결과값은?

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 10;

    printf("b >= a: %d\n", b >= a);

    return 0;
}
```



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 논리

- $A \ \&\& \ B$: A 그리고 B(A and B)
- $A \ || \ B$: A 또는 B(A or B)
- $!A$: A가 true(1)일 때 false(0)반환, false(0)일 때 true(1) 반환



연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 조건(삼항)

- $A ? B : C$: A가 true(1)일 때 B 반환
false(0)일 때 C 반환

연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 조건(삼항)

- 변수 c의 값은?

```
#include <stdio.h>
int main() {
    int a = 3;
    int b = 10;
    int c;

    c = b > a ? 1 : 0;

    printf("c: %d\n", c);

    return 0;
}
```

연산자

산술

할당

증감

관계

논리

조건

우선 순위

| 연산자 우선 순위

우선순위	연산자	결합수칙
1	() [] . ->	왼쪽 우선
2	*(간접지정) & ! ++ --	오른쪽 우선
3	*(곱셈) % /	왼쪽 우선
4	+ -	왼쪽 우선
5	<< >>	왼쪽 우선
6	< > <= >=	왼쪽 우선
7	== !=	왼쪽 우선
8	&	왼쪽 우선
9	^	왼쪽 우선
10		왼쪽 우선
11	&&	왼쪽 우선
12		왼쪽 우선
13	? :	오른쪽 우선
14	= += -= *= %= /= ^= <<= >>=	오른쪽 우선
15	,	왼쪽 우선

6. 기타



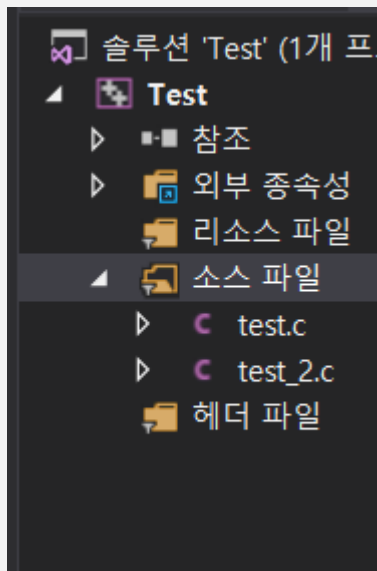
기타

소스 파일 다중 생성

LNK 오류

이 시스템 주의사항

| 소스 파일 다중 생성



- 2개 이상의 소스 파일 생성 시
실행하면 **빌드 오류 발생**



기타

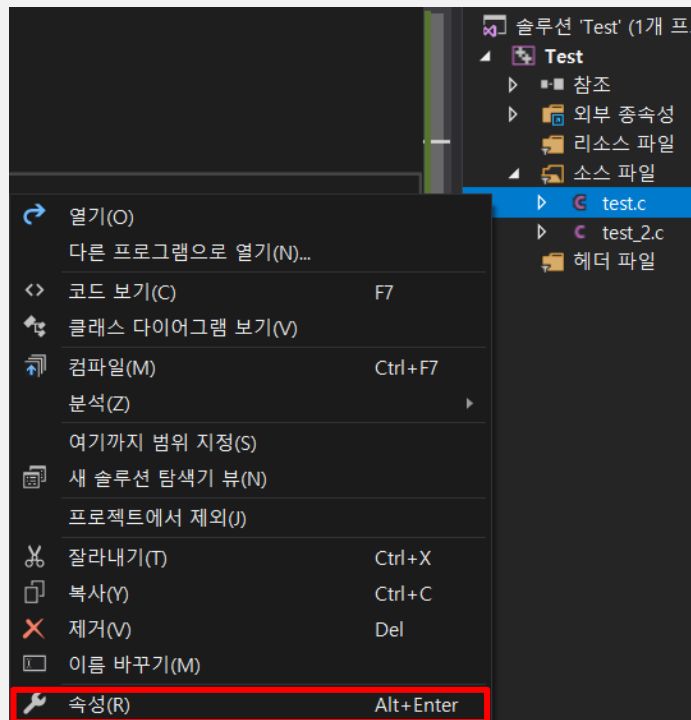
소스 파일 다중 생성

LNK 오류

이 시스템 주의사항

| 소스 파일 다중 생성

- 실행하지 않을 소스 파일 > 속성





기타

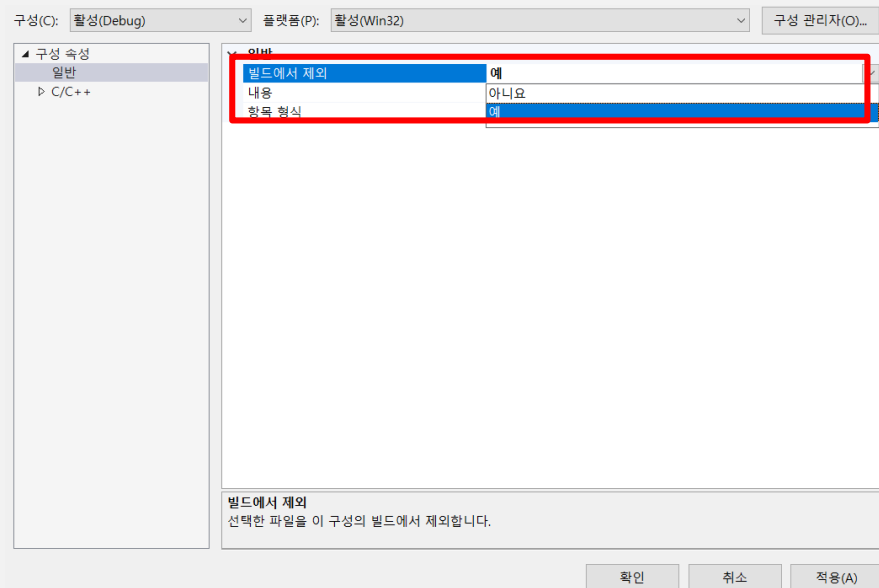
소스 파일 다중 생성

LNK 오류

이 시스템 주의사항

| 소스 파일 다중 생성

- 빌드에서 제외 > 예





기타

소스 파일 다중 생성

LNK 오류

이 시스템 주의사항

| LNK 1120 오류 해결 방법

- main함수가 제대로 생성되었는지 확인하기
- 잘못된 함수를 사용했는지 확인하기
- 소스 파일을 .c가 아닌 .cpp로 만들었는지 확인하기
- 위 경우에 모두 해당하지 않는다면, 옳은 방법으로 프로젝트 다시 만들기



기타

소스 파일 다중 생성

LNK 오류

이 시스템 주의사항

I O 시스템 사용 시 주의사항

- 문제의 출력 예시와 자신의 코드 출력 비교
 - 공백(' ') 혹은 개행('\n')문자 등으로 출력이 다르면 X
- 런타임 에러 && 컴파일 에러
 - 대부분 문법 오류(문장 끝에 ';' 부재, 스펠링 틀림)
- 타임리미트 에러
 - 반복문 탈출 조건이 없는 등의 이유로 실행시간 길어짐