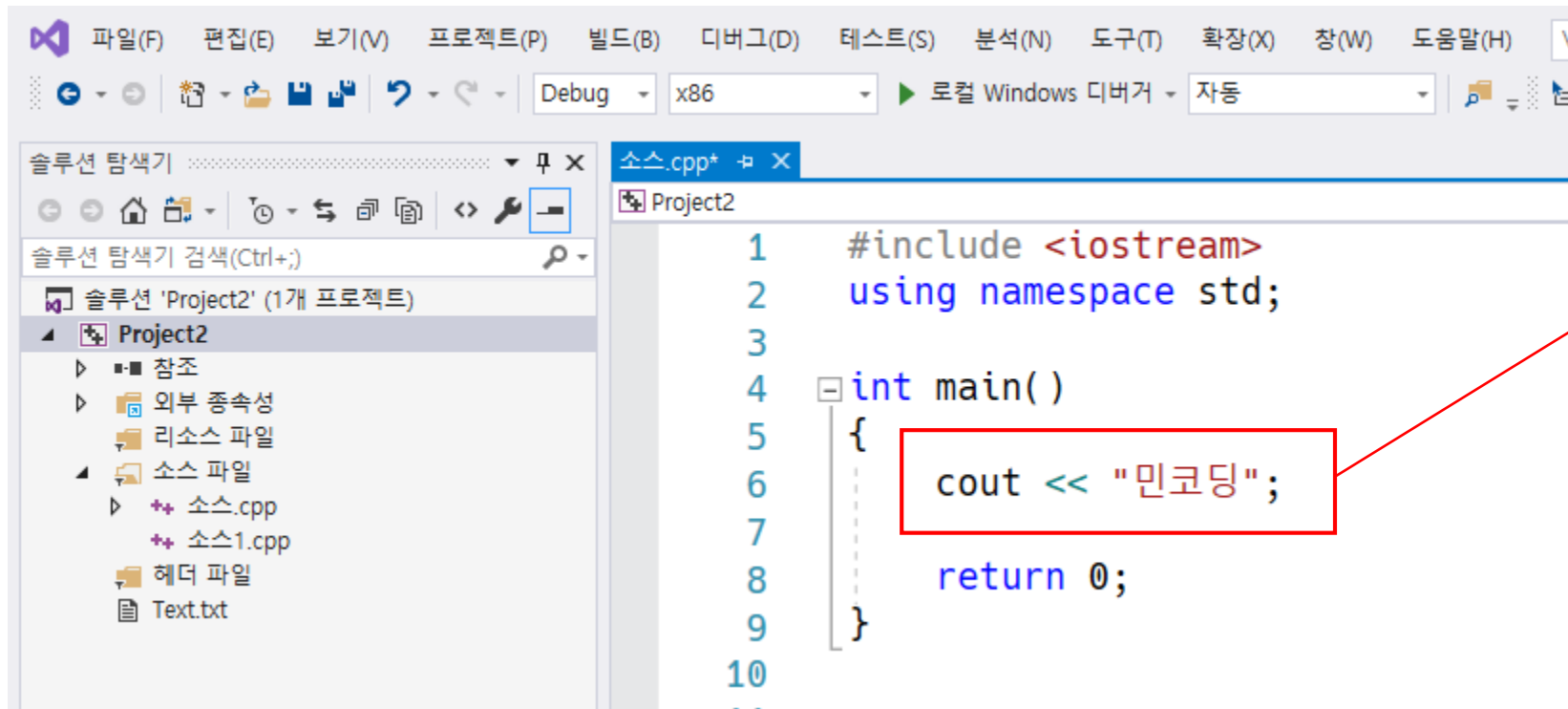


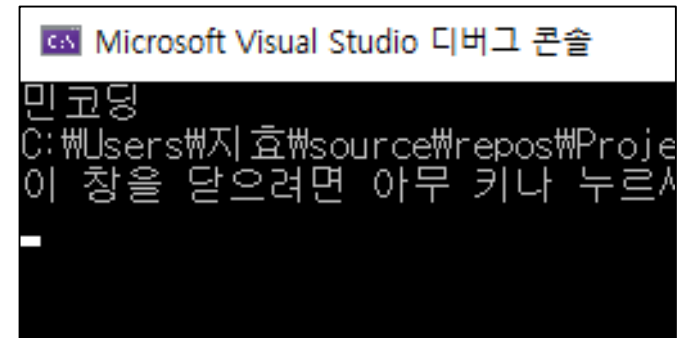
배우는 내용

1. cout
2. 변수
3. trace(트레이스)

글자 출력하는 소스코드



민코딩이라는 글씨를 출력함
Ctrl + F5를 눌러서 빌드 (결과확인 가능)



빌드 결과 확인 (Ctrl + F5)

*cout : 씨아웃이라고 말합니다. Console Output의 약자 입니다

The screenshot shows the Visual Studio IDE with a C++ project named 'Project2'. The source file '소스.cpp' contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "하하하";
6
7     return 0;
8 }
9
10
11
```

The 'Output' window at the bottom left shows the build process:

```
출력
출력 보기 선택(S): 빌드
1>----- 빌드 시작: 프로젝트: Project2, 구성: Debug Win32 -----
1>소스.cpp
1>Project2.vcxproj -> C:\Users\zz\호\source\repos\Project2\Debug\Project2.exe
----- 빌드: 성공 1, 실패 0, 최신 0, 생략 0 -----
|
```

Below the output, a red box highlights the message: **빌드에 성공했음을 나타냄** (Indicates successful build).

At the bottom, a status bar message says: **빌드에 성공했습니다.** (Build successful).

On the right, a red box highlights the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window, which displays the program's output:

```
하하하
C:\Users\zz\호\source\repos\Project2\Debug\Project2.exe(14516 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

Red text above this window states: **결과화면 = 콘솔창이라고 함** (Result screen = console window).

기본 소스코드의 해석

```
#include <iostream>

int main( )
{
    std::cout << "민코딩";

    return 0;
}
```

iostream.h 파일을 불러와야
std::cout 명령어를 사용할 수 있음

```
#include <iostream>

int main( )
{
    std::cout << "민코딩";

    return 0;
}
```

std::cout은 민코딩 글자를 출력한다


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

int main( )
{
    cout << "민코딩";

    return 0;
}
```

std::cout에서
std:: 글자를 생략하기 위해
using namespace std; 를 적어준다.

코딩 공부를 위한 기본 소스코드

```
#include <iostream>
using namespace std;
int main()
{
    
    return 0;
}
```

이 사이에 소스코드를 넣으면 된다
공백이 많이 들어가도 상관없음

이 형틀을 잊어버리지 않도록 반복 연습을 통해 기억하자.
프로그래밍 공부의 가장 기본이 되는 소스코드

cout 의 특징

두 소스코드는 같은 결과가 나온다
마음에 드는 방식으로 코딩

```
#include <iostream>
using namespace std;
int main()
{
    cout << "민코딩";
    cout << "개발자";
    cout << "BBQ";

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    cout << "민코딩개발자BBQ";

    return 0;
}
```

빌드

Microsoft Visual Studio 디버그 콘솔

민코딩개발자BBQ
C:\Users\지효\source\repos\Project2\Deb
이 창을 닫으려면 아무 키나 누르세요.

엔터 출력

다음과 같이 출력하고 싶다면?

민코딩
개발자
BBQ

```
#include <iostream>
using namespace std;
int main( )
{
    cout << "민코딩" << endl;
    cout << "개발자" << endl;
    cout << "BBQ" << endl;

    return 0;
}
```

엔터를 뜻함

Microsoft Visual Studio 디버그 콘솔

민코딩
개발자
BBQ

C:\Users\지효\source\repos\Project
이 창을 닫으려면 아무 키나 누르세

변수란?

- 숫자나 글자 같은 값을 컴퓨터 안에 저장하는 박스
- 박스를 만들 때 이름을 정해줘야 함
- 변수(박스)의 종류 예시



숫자(정수)가 들어가는 int 타입 박스

=인트형 변수



글자가 들어가는 char 타입 변수

=캐릭터형 변수



소수점 숫자가 들어가는 double 타입 박스

=더블형 변수

변수 만드는 법

a라는 박스를 만들고 박스 안에 10을 넣었음
둘 중 마음에 드는 방식으로 하면 된다

a

10

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 10;

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;

    return 0;
}
```

위 소스코드를 빌드 후 실행결과를 보면 화면에 아무것도 뜨지 않습니다.
박스를 만들었을 뿐, cout 명령어로 글자를 출력하지 않았기 때문입니다

두 가지 알아야 할 전문용어

- a라는 이름의 변수(박스)를 만든다
→ 변수 a를 선언한다
- 변수 a에 값 10을 넣는다
→ 변수 a를 10으로 초기화한다

변수 3개 선언하고 각각 1, 2, 3으로 초기화하기

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    int b;
    int c;

    a = 1;
    b = 2;
    c = 3;

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    a = 1;

    int b;
    b = 2;

    int c;
    c = 3;

    return 0;
}
```

버그 발생 코드1

버그 : 벌레, 프로그래머 실수로 인한 오류를 뜻함

왜 버그일까요?

→ 박스를 먼저 만들고 안에 값을 넣어야 하기 때문에 버그 발생

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    int b;
    b = 2;
    c = 3;
    ~int c;

    return 0;
}
```

버그 발생 코드2

왜 버그일까요?

→ 같은 이름의 변수를 2개 만들었기 때문

```
#include <iostream>
using namespace std;
int main()
{
    int a = 40;
    int b = 20;
    int c = -5;

    a = 15;
    b = 22;
    int c = 15;

    return 0;
}
```

변수 만드는 법

```
int a;           // a라는 변수(박스)를 만들었다.  
int bbq123;      // bbq123이라는 변수를 만들었다.  
int a, b, c;     // a, b, c라는 변수를 만들었다.  
int d1 = 5, d2 = 10; //d1과 d2 변수를 만들자마자 값을 5와 10을 넣었다.
```

- 주의 : 변수명은 숫자로 시작할 수 없음 (Ex. int 123abc)
- 선언된 변수 안에는 초기값으로 쓰레기값이 자동으로 들어감
쓰레기값 : 의도하지 않은 값. 아무 값도 넣지 않았을때 자동으로 넣어지는 값

변수의 활용1

```
int a =10;
```

```
a = 100;
```

```
// 변수 a에 10을넣고, 다시 a값을 100으로 바꾸었다.
```

```
// a 안에 있던 10은 없어지고, 값 100이 들어가 있다.
```

```
int g = 1, k = 2;
```

```
int c = g + k;
```

```
//c라는 변수에는 숫자 3이 들어간다
```

변수의 활용2

```
int g;
```

```
g = 10 / 3;
```

```
//변수 g에는 숫자 3.3333 이 들어있지않고 숫자 3이 들어있다.
```

```
//int 소수점을 저장하는 박스가 아니라, 정수를 저장하는 박스이다.
```

```
//따라서 소수점 뒷자리를 버리고 숫자 3이 들어간다.
```

```
int a = 3 * 10;
```

```
//a에는 숫자 30 이 들어있다.
```


변수 용어정리

- `int a;` //변수 `a`를 "선언"한다 라고 표현
- `int a = 10;` //변수 `a`를 선언 후 10으로 "초기화" 한다 라고 표현
- `int a = 1, b = 10;` //변수 `a, b` 선언 후 각각 1과 10으로 초기화 한다 라고 말한다.

Trace 하기

단축키를 외워주세요

- 빌드 (Ctrl + F5) : 모든 소스코드들을 한번에 실행하기
- Trace (F10) : 소스코드 한 줄씩 실행하기
- Trace 끄기 단축키 : Shift + F5

```
#include <iostream>
using namespace std;
int main()
{
    cout << "하하" << endl;
    cout << "호호" << endl;
    cout << "완료";

    return 0;
}
```

빌드

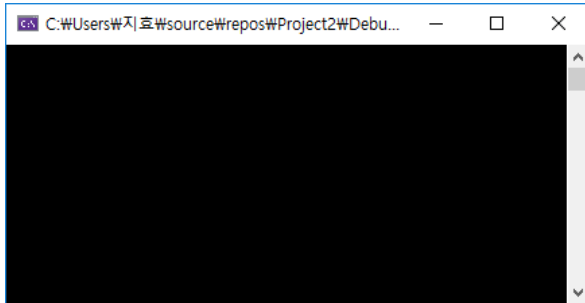
트레이스

노란색 화살표가 등장

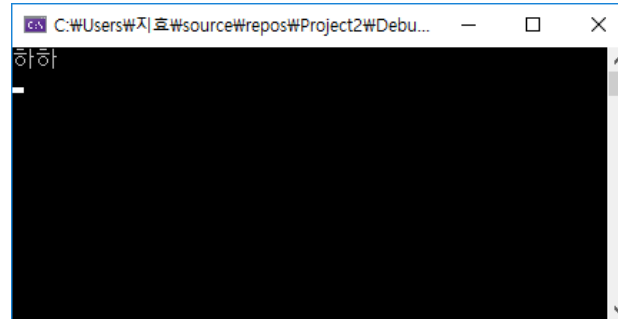
Trace 하기

- F10을 반복적으로 누르면 한줄 씩 실행된다
- 종료하는 단축키는 Shift + F5

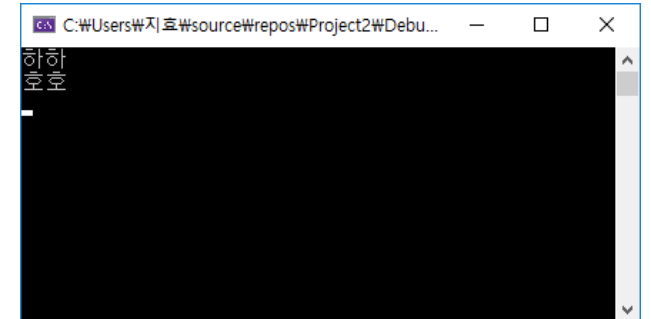
```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "하하" << endl;
6      cout << "호호" << endl;
7      cout << "완료";
8
9      return 0;
10 }
```



```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "하하" << endl;
6      cout << "호호" << endl;
7      cout << "완료";
8
9      return 0;
10 }
```



```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << "하하" << endl;
6      cout << "호호" << endl;
7      cout << "완료"; 경과 시간 41
8
9      return 0;
10 }
```

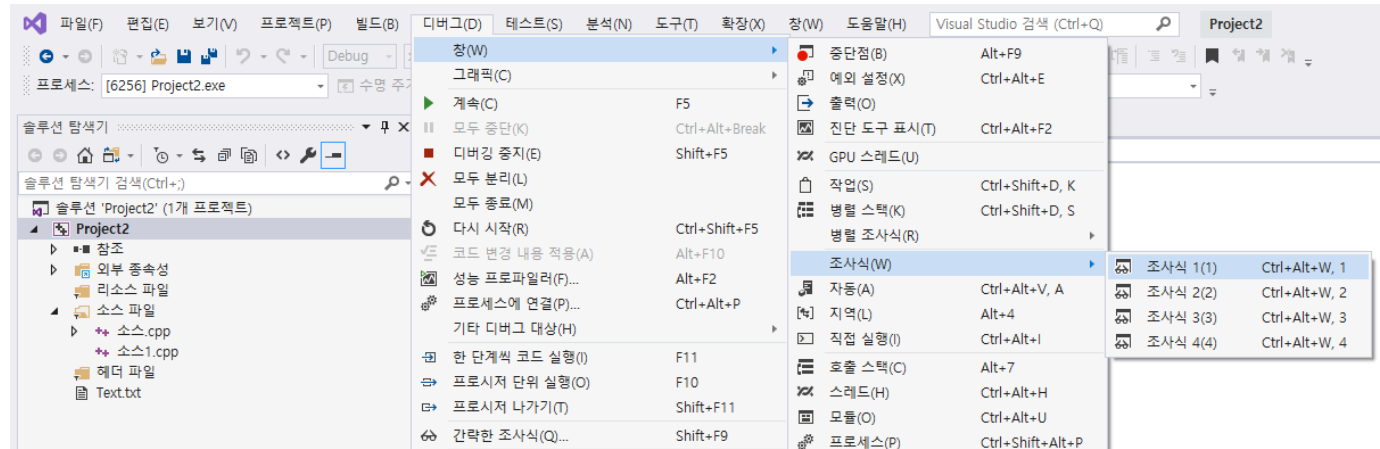


변수에 들어가 있는 값 확인하기1

- Trace로 변수에 어떤 값이 들어있는지 확인할 수 있음
- 아래 소스코드를 입력하고 F10 누르기
- 그리고 [디버그] > [창] > [조사식] > [조사식1] 을 눌러서 조사식 창을 열면 됨

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    return 0;
}
```

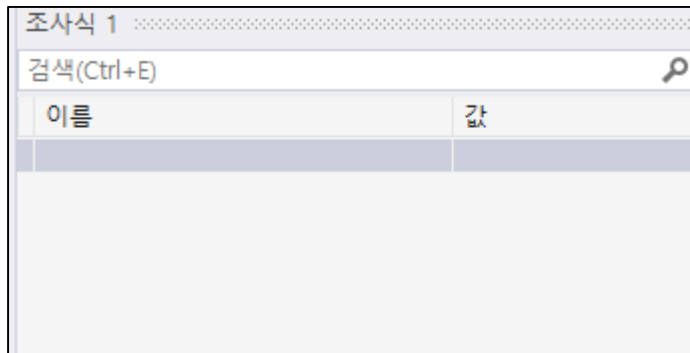


변수에 들어가 있는 값 확인하기2

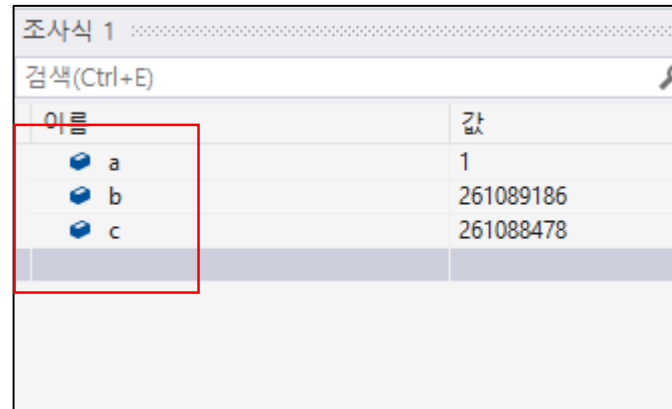
- 조사식에 변수 a, b, c 값을 확인하기 위해 a, b, c를 등록함
- 그리고 F10을 반복적으로 눌러보면 a, b, c에 값이 들어가는 것을 확인할 수 있음

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

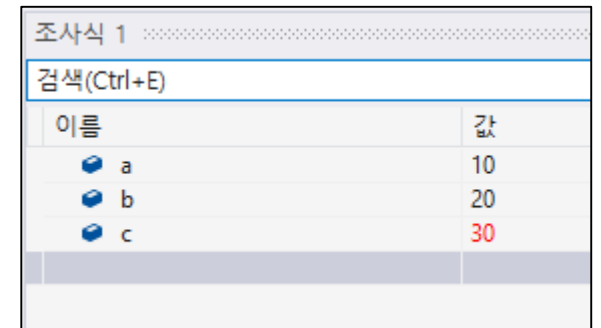
    return 0;
}
```



변수 이름 a,b,c 등록하기



이제 F10을 천천히 반복적으로 눌러봄



변수 a, b, c 값이 들어감

Trace를 하는 이유

- 이 기능으로 버그를 찾아낼 수 있음 = 버그를 죽이는 행동 = 디버깅 이라고 한다
빠른 디버깅을 위해 Trace를 사용한다.
- 소스코드가 복잡한 경우 Trace없이 버그 찾기 어려움
- 프로그래머는 디버깅 능력을 반드시 키워야 한다
소스코드를 작성 후 Trace를 한번씩 해 보는 것을 권장 함 (강력추천)

변수 값 출력하기

cout을 이용해서 변수의 값을 화면에 출력할 수 있음
글자를 출력하는 것과 헷갈리지 않도록 주의

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << a;

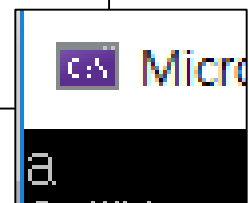
    return 0;
}
```



```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << "a";

    return 0;
}
```



글자와 변수를 동시에 출력하기1

<< 를 여러 번 써서 글자와 변수를 동시에 출력 가능

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << "a의 값은 : " << a << "입니다" << endl;

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

a의 값은 : 10입니다

글자와 변수를 동시에 출력하기2

<< 를 여러 번 써서 글자와 변수를 동시에 출력 가능

```
#include <iostream>
using namespace std;
int main()
{
    int a = 10;
    int b = 20;
    int c = a + b;

    cout << a << "와 " << b << "를 더하면";
    cout << c << "입니다";

    return 0;
}
```

 Microsoft Visual Studio 디버그 콘솔

10와 20를 더하면30입니다

C언어 공부할 때 자주 발생하는 버그

- ▶ 버그 : 문장 끝에 **세미콜론(:)**을 붙이지 않는 경우
→ 세미콜론 안 붙인 곳 **주변에** 빨간 밑줄이 그어집니다.

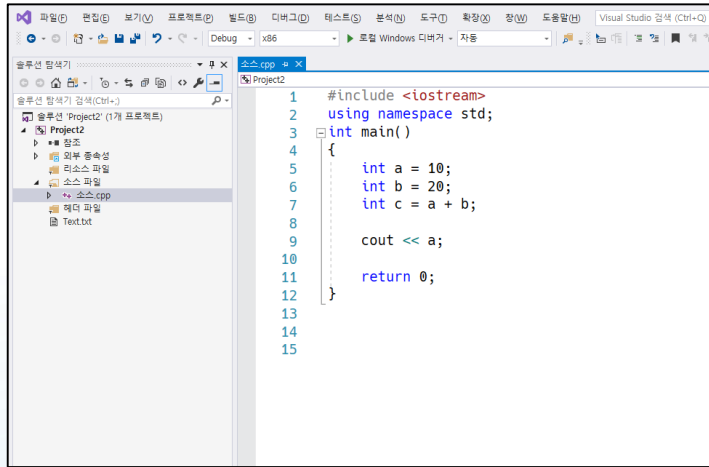
```
#include <iostream>
using namespace std;
int main( )
{
    int a = 10;
    int b = 20;
    int c = a + b

    return 0;
}
```

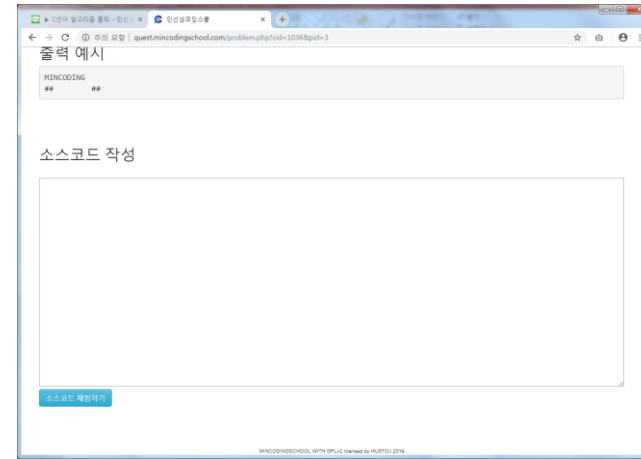
이곳에서 버그 발생 ! 세미콜론을 붙이지 않음

버그 문장 주변에 빨간 밑줄이 그어짐
(이곳은 버그가 아님)

숙제하실 때 주의할 점 두가지




복사 붙여넣기



코딩은 Visual Studio에서 하시고, 빌드 까지 해서 오류가 없는지 확인 하세요
그리고 완료가 되었으면 소스코드를 복사 붙여넣기를 해서 제출하시면 됩니다.

하나. Visual Studio에서 코딩하세요 (훈련사이트에서 코딩하지마세요)
둘. 모든 숙제는 기존 소스코드를 지우고, 기본 코드부터 시작해주세요



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

int main()
{

    return 0;
}
```