

2023-1 인트아이 C++ 수업 자료

C++ 기초 1강 입출력





CONTENTS

01. 프로젝트 생성

02. 입출력

03. 깃허브로 push 하기

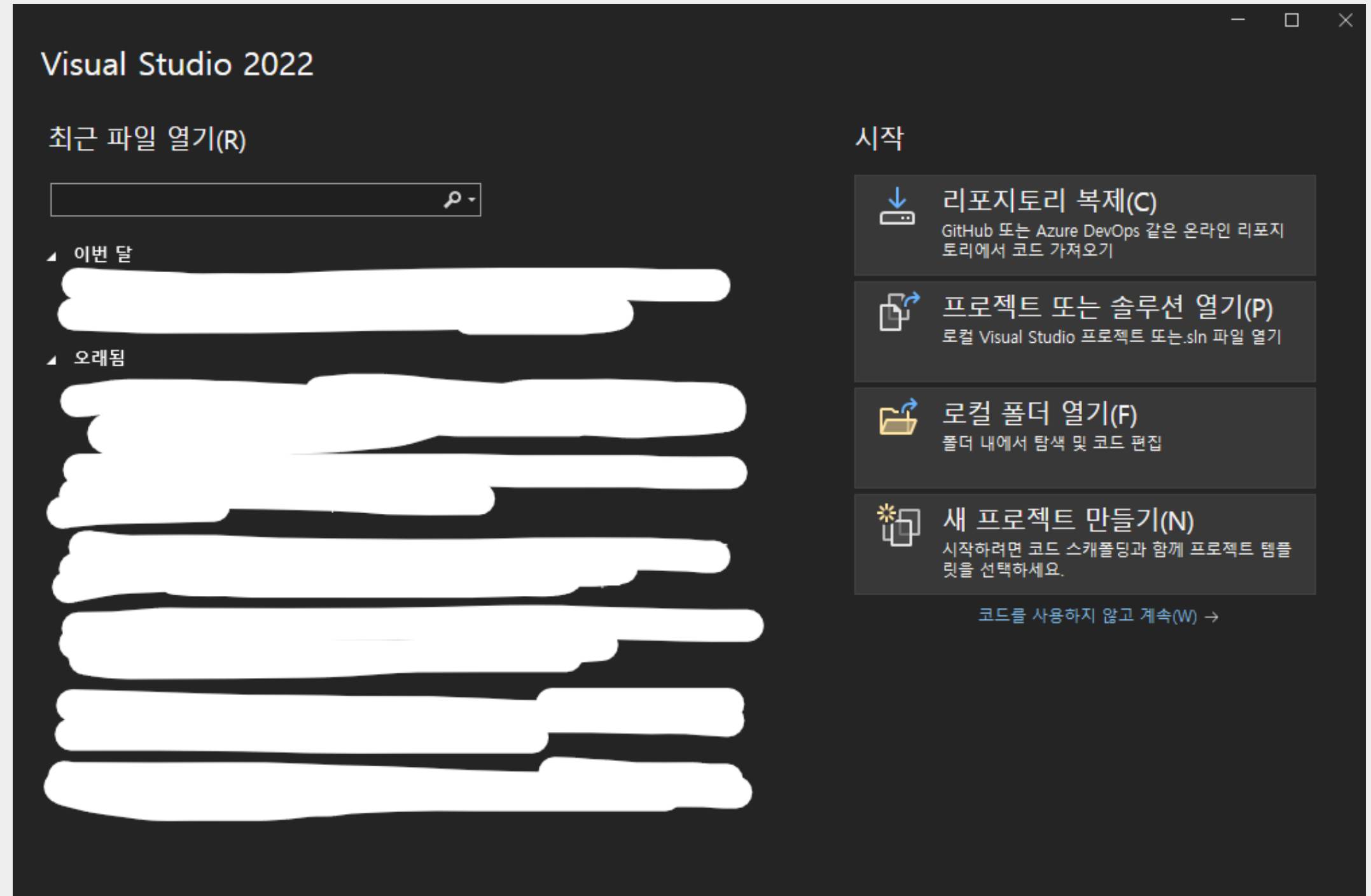
우선은 깃허브부터 들르자

https://github.com/int-i/2023-1-cpp-study 에 접속해보자.
해당 리포지토리가 앞으로 과제를 올리게 될 리포지토리이다.
아래 화면에 보이는 대로 리포지토리 복제를 위한 링크를 복사해주자

The screenshot shows the GitHub interface for the repository 'int-i / 2023-1-cpp-study'. The repository is public and has 1 branch (main) and 0 tags. The 'Code' dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI. The HTTPS URL 'https://github.com/int-i/2023-1-cpp-study.git' is highlighted and a 'Copied!' notification is shown. Other options include 'Open with GitHub Desktop', 'Open with Visual Studio', and 'Download ZIP'. The right sidebar shows the repository's 'About' section, indicating 0 stars, 1 watching, and 0 forks. There are also sections for 'Releases' and 'Packages'.

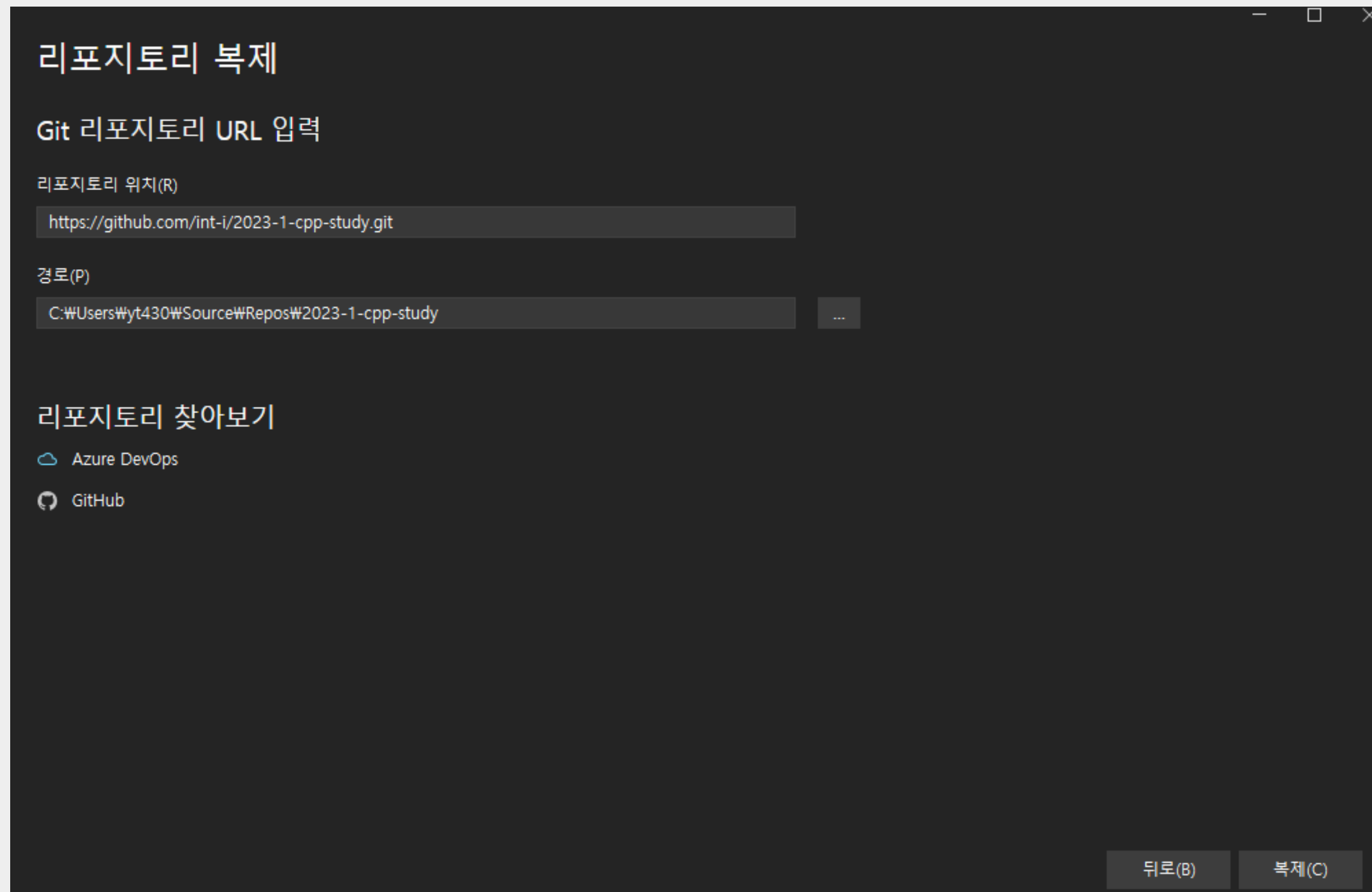
Visual Studio를 켜주자

우리가 방금 보았던 그 리포지토리를 복제하기 위해
리포지토리 복제를 눌러주자



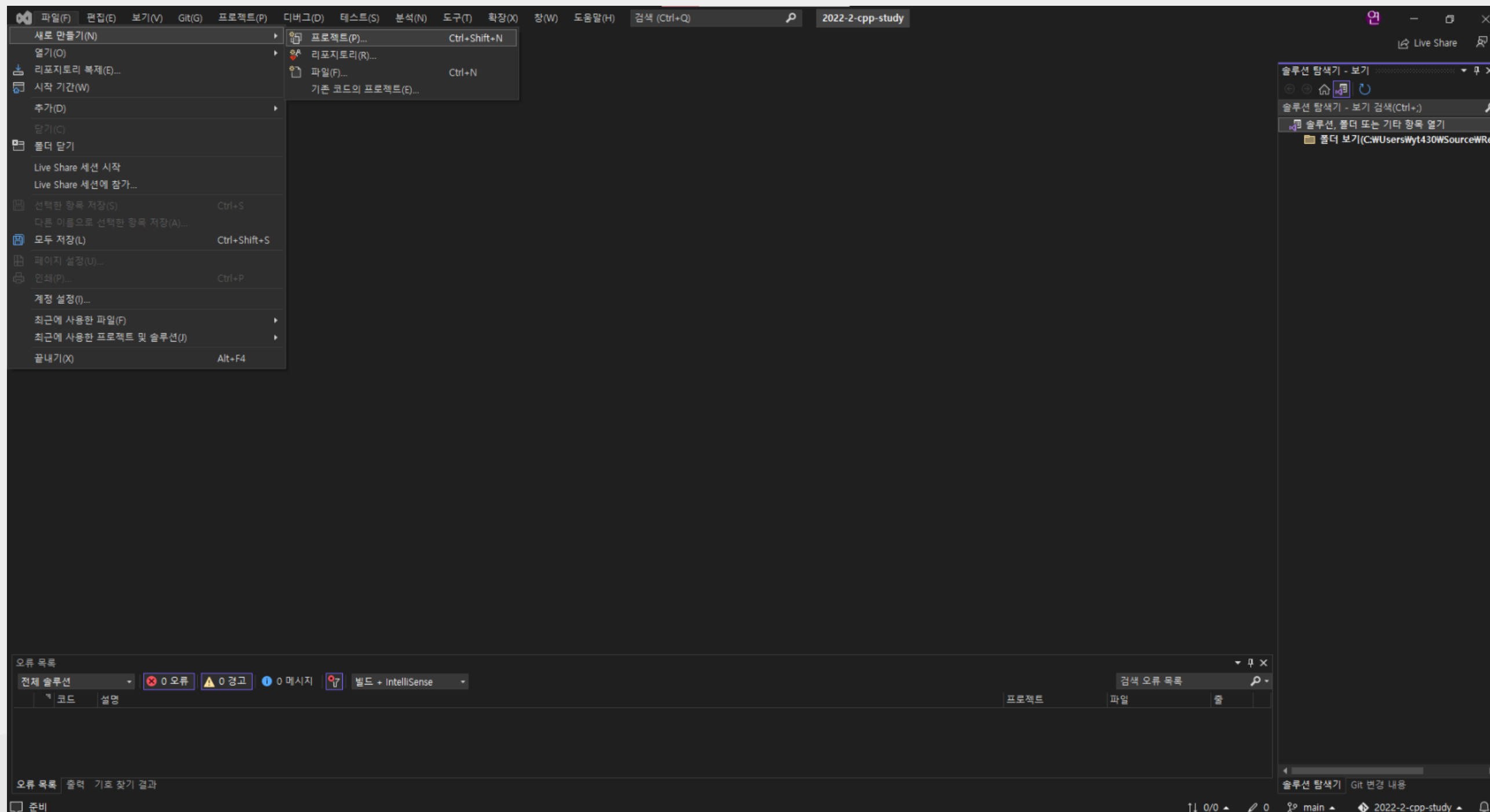
리포 복제 진행

그 다음 리포지토리 위치에 아까 복사했던 그 링크를 붙여넣고,
(ssh가 안되면 https로 진행) 복제를 눌러준다.



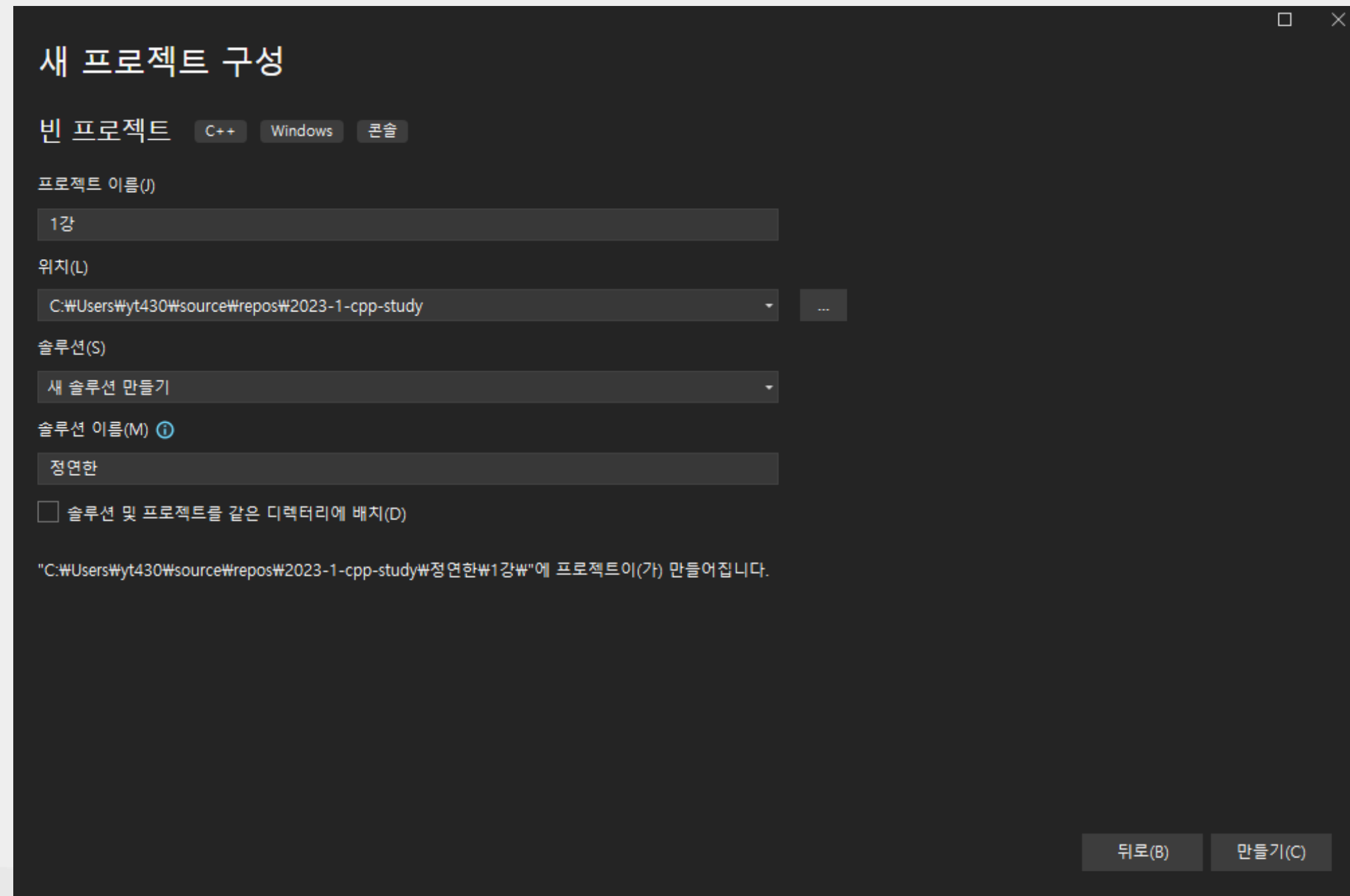
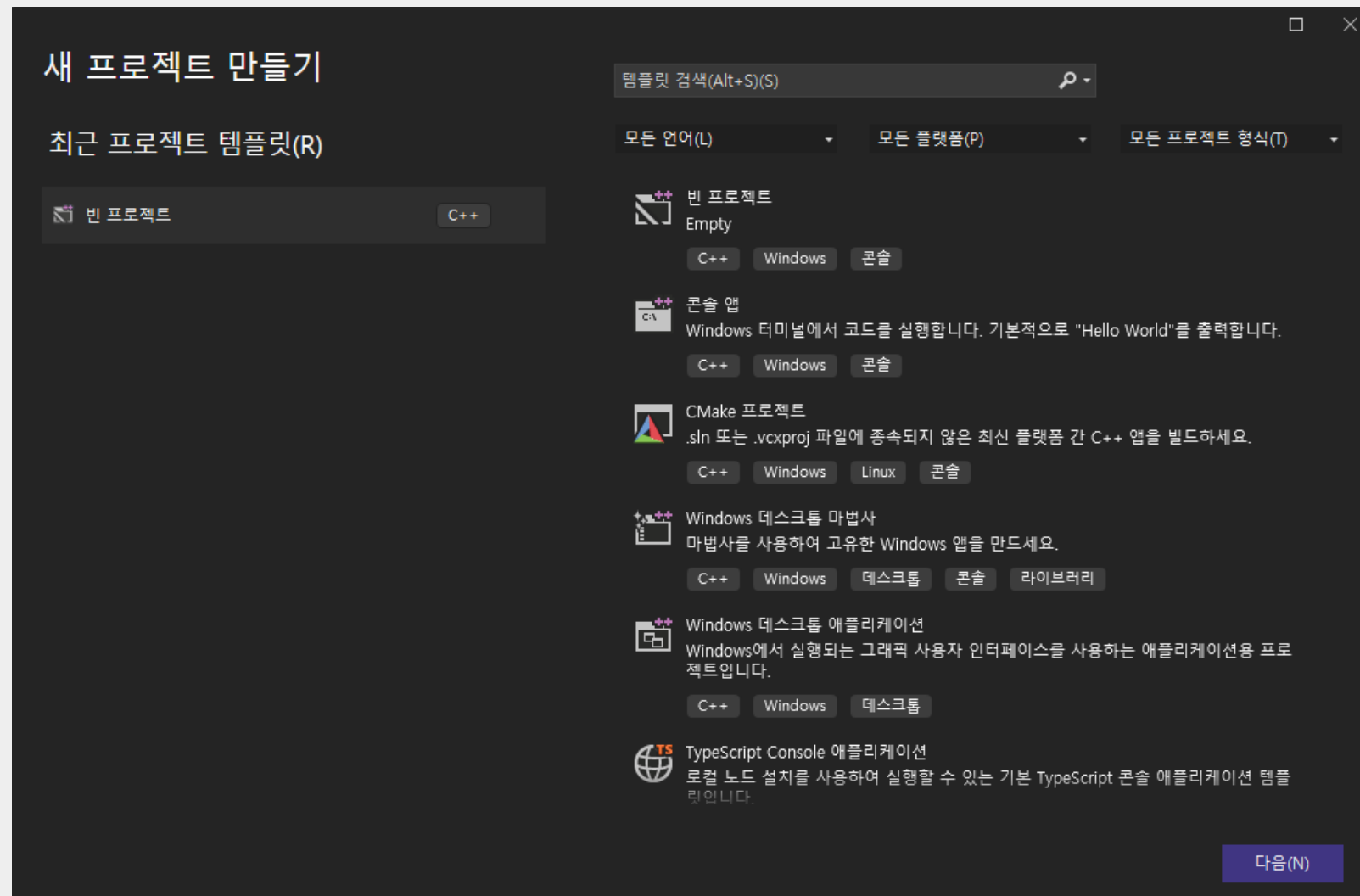
리포 복제 진행

복제가 완료되면 아래와 같이 솔루션 탐색기에 폴더 보기만 뜨고 아무것도 뜨질 않는다.
여기서 파일 - 새로 만들기 - 프로젝트를 눌러준다.



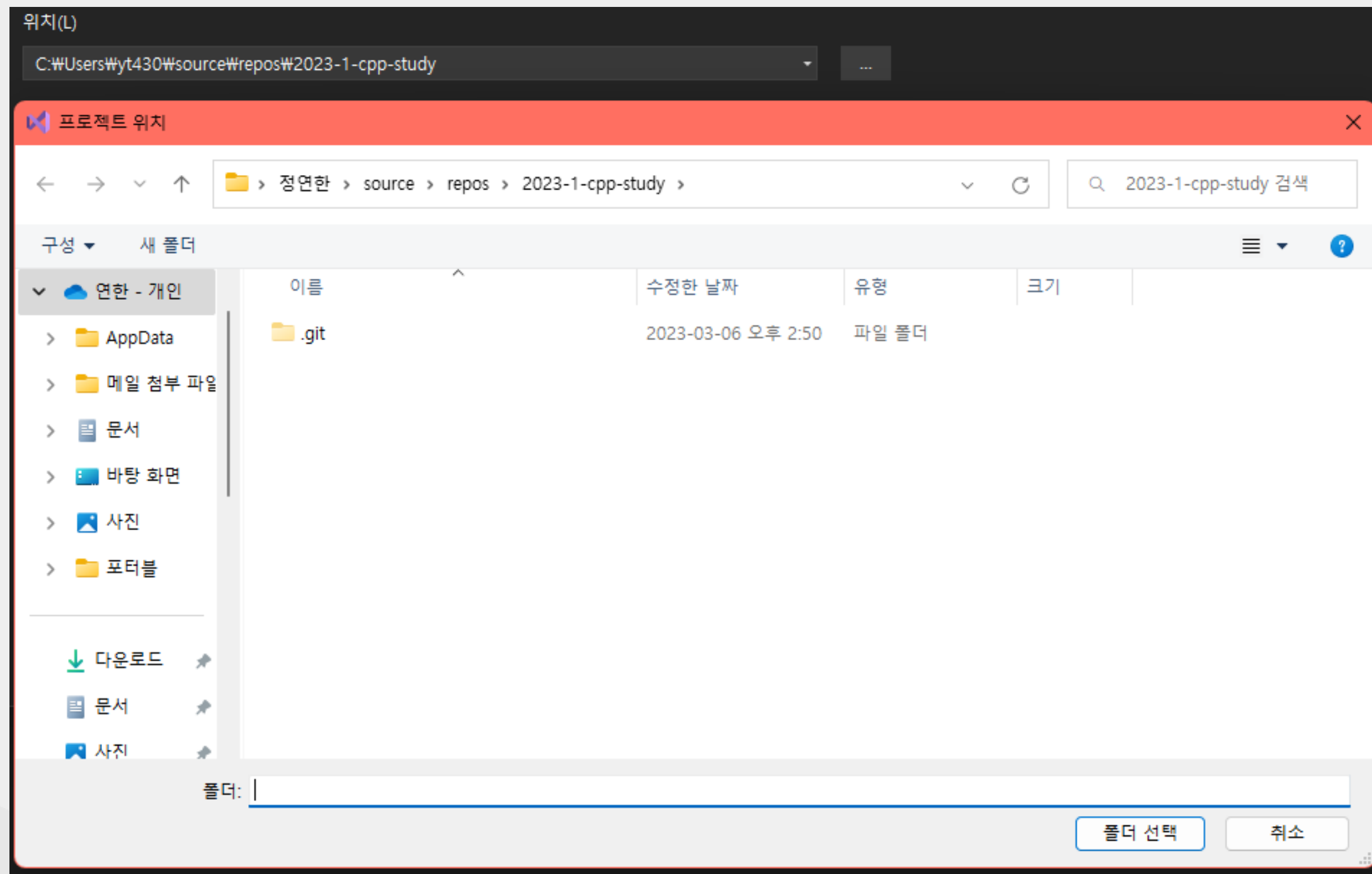
새 프로젝트 구성

기존에 하던대로 새 프로젝트를 만드는데, 이때 주의해야 할 것이 몇가지가 있다.
우선 프로젝트 이름은 {n}강 으로, 위치는 방금 전에 복제했던 리포 안으로,
솔루션 이름은 본인 이름으로 해주어야 한다.
원활한 프로젝트 관리를 위한 것이므로 반드시 동일하게 진행 할 것!



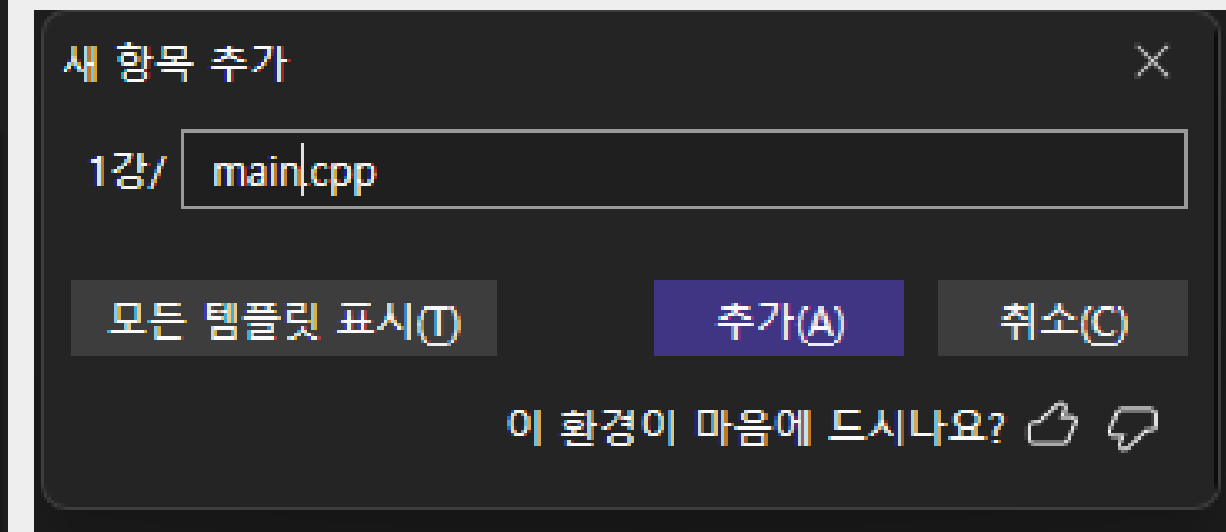
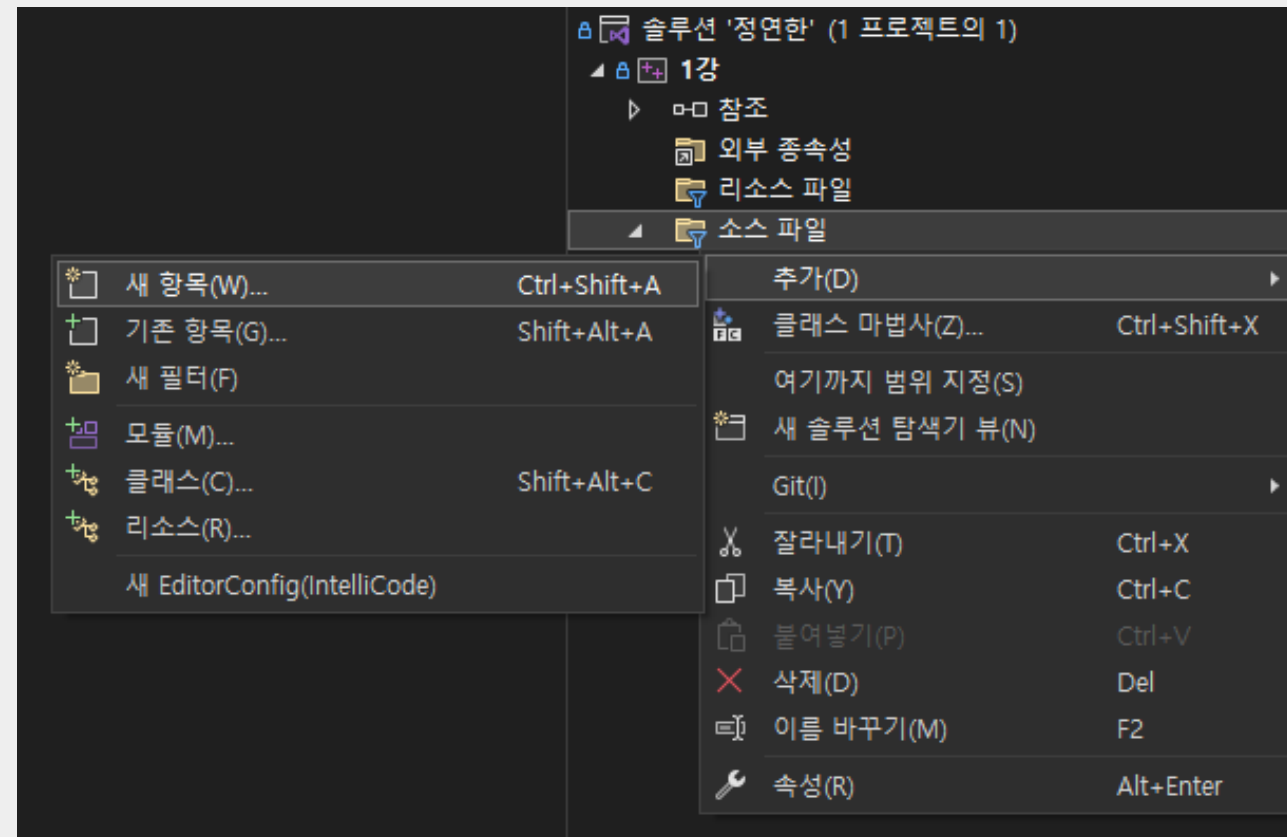
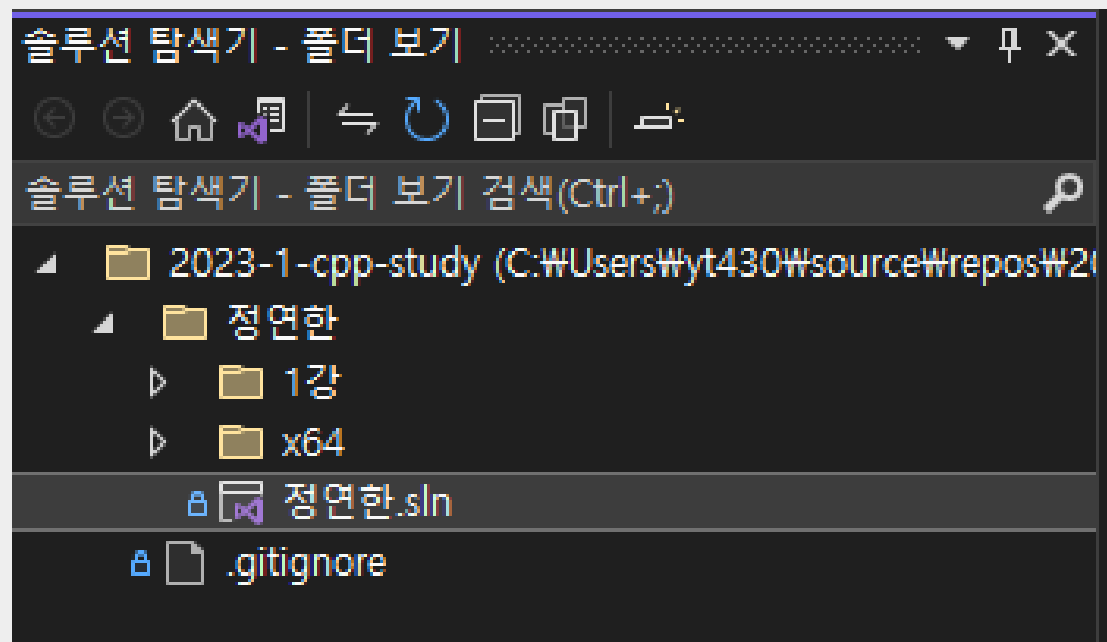
새 프로젝트 구성

혹시나 위치 구성이 잘못되어 있을 경우 아래의 사진을 보고 위치를 다시 잡아줘야 한다!
새 프로젝트의 위치는 반드시 "깃허브에서 복제"한 "2023-1-cpp-study" 폴더를 선택할 것!!



새 cpp 파일 생성

자, 여기까지 했다면 프로젝트 생성은 완료다!
이제 방금 만든 자신의 이름으로 된 폴더를 열어보면 {본인이름}.sln이 있을 텐데
그걸 열어주면 우리가 만든 솔루션을 열 수 있다!
이제 여기서 소스 파일 마우스 오른쪽 클릭 -> 추가 -> 새 항목
을 통해 새 소스 파일을 생성해주자.



C++ 코드 작성

간단한 코드를 작성해보자. 주석은 일단 신경 쓰지 말고 주석이 아닌 코드만 작성해주자.

주석이 뭐냐고? // 뒤에 작성된 부분을 주석이라 한다.

주석은 컴파일러가 읽지 못하기 때문에 프로그래머끼리의 소통을 위해 작성이 된다.

다른 동료 프로그래머가 될 수도 있고, 미래의 내가 될 수도 있고

```

1 // input, output stream의 약자
2 // 말 그대로 입출력 스트림과 관련된 헤더파일이다.
3 // 스트림이란 운영체제에 의해 생성되는 가상의 연결 고리를 의미하며, 중간 매개자 역할을 한다.
4 #include <iostream>
5
6 // std라는 네임스페이스를 써라 하고 컴파일러에게 지시하는 것
7 // 그래서 이름도 using 지시문
8 using namespace std;
9
10 // C++ 프로그램의 시작이자 끝을 담당하는 main 함수
11 // C++ 프로그램이 시작되면 main 함수가 시작되고,
12 // main 함수가 종료되면 C++ 프로그램이 종료된다.
13 int main() {
14     // 출력은 cout을 통해 할 수 있다.
15     // cout 뒤에 <<(출력 스트림 연산자)를 통해 출력하고자 하는 값을 작성해주자.
16     cout << "Hello inti!" << endl;
17
18     // 함수를 종료하는 return 문, 이 줄이 실행이 되면 main함수가 종료된다.
19     return 0;
20 }
21

```

#include <iostream>이란?

자 첫 줄 부터 이상한게 나왔다. 샵 인클루드 아이오스트림?

#은 전처리기로, 내가 짠 코드를 실행시킬 수 있는 형태로 만들기 전 실행하라고 명령하는 부분이다.

include는 말 그대로 포함하라는 뜻이다. 무엇을? 오른쪽에 온 파일을.

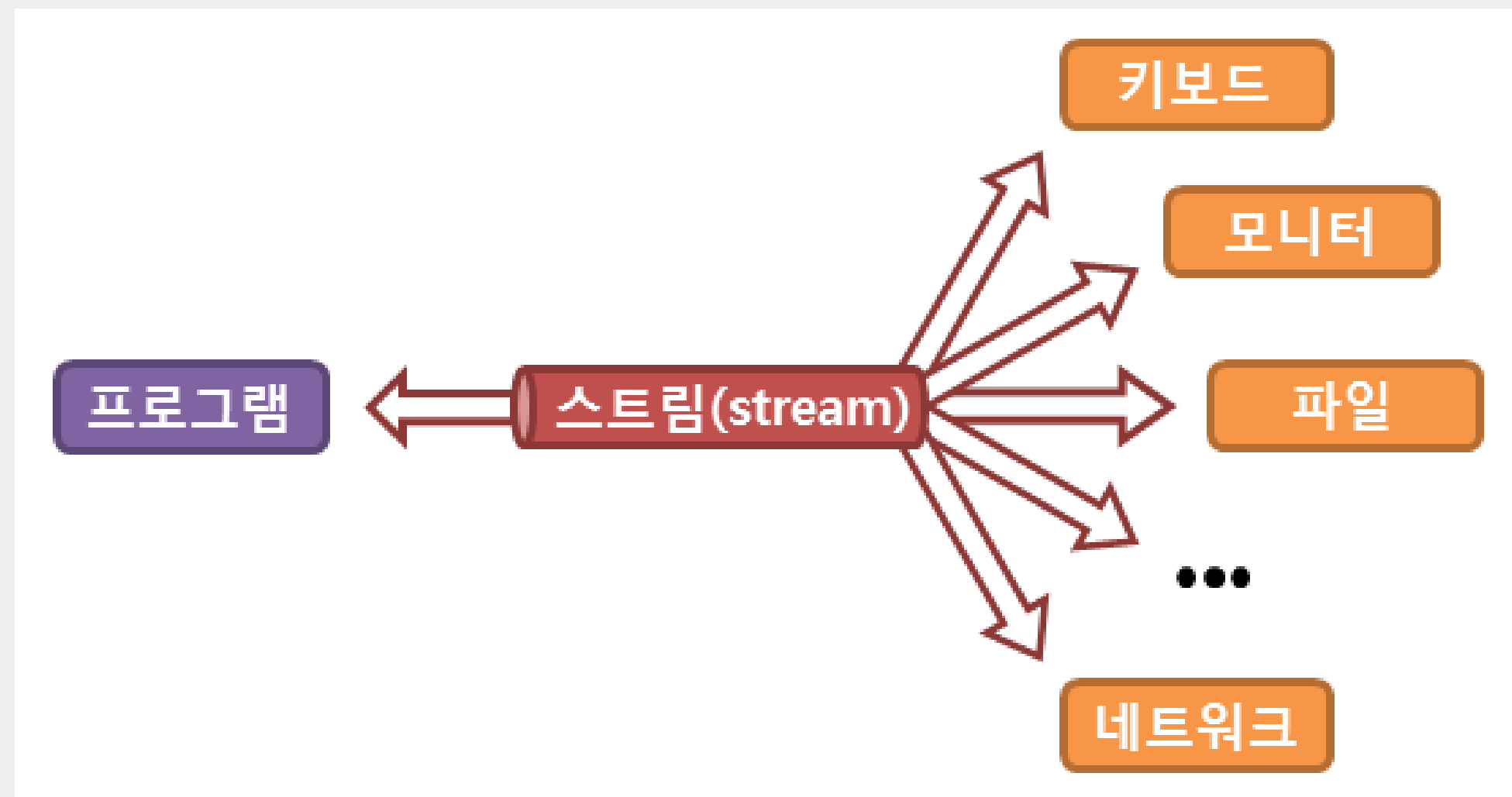
iostream은 입출력 스트림이 선언된 헤더파일이다.

즉 다시 해석하자면 입출력을 위해 iostream이라는 파일을 이 코드에 불러오겠다. 라는 뜻이 되겠다.

```
// input, output stream의 약자  
// 말 그대로 입출력 스트림과 관련된 헤더파일이다.  
// 스트림이란 운영체제에 의해 생성되는 가상의 연결 고리를 의미하며, 중간 매개자 역할을 한다.  
#include <iostream>
```

스트림이란?

C++ 프로그램은 파일이나 콘솔의 입출력을 직접 다루지 않고, 스트림(stream)이라는 흐름을 통해 다루게 되는데, 이때 스트림(stream)이란 실제의 입력이나 출력이 표현된 데이터의 이상화된 흐름을 의미한다.
즉, 스트림은 운영체제에 의해 생성되는 가상의 연결 고리를 의미하며, 중간 매개자 역할을 한다.



namespace란?

네임스페이스는 모든 식별자(변수, 함수, 형식 등의 이름)가 고유하도록 보장하는 코드 영역을 정의한다.
고작 한 줄인데, 딱 보자마자 이해가 되진 않아보인다.

우선 식별자란 말 그대로 어떠한 개체를 다른 개체와 달리 식별할 수 있게 하는 것을 의미한다.
변수명, 함수명 등 특정 개체의 이름이 이에 해당한다.

즉, 네임스페이스란 내가 만든 변수명이 다른 변수명과 충돌나지 않도록(고유하도록) 해주는 코드 영역
지금은 잘 이해가 안되겠지만, 나중에 다시 와서 읽어보면 아, 이런 내용이구나 싶을 것이다.

```
// std라는 네임스페이스를 써라 하고 컴파일러에게 지시하는 것  
// 그래서 이름도 using 지시문  
using namespace std;
```

main 함수란?

main 함수는 C++ 프로그램의 시작이자 끝이다.

C++ 프로그램이 실행이 되면, main 함수가 시작되고,

main 함수가 종료되면 C++ 프로그램이 종료가 된다.

가끔 어떤 자료를 보면 void main() 이라고 쓰는 경우도 있는데, 비표준이다. 그렇게 쓰지 말자.

표준은 int main()이다.

```
10 // C++ 프로그램의 시작이자 끝을 담당하는 main 함수
11 // C++ 프로그램이 시작되면 main 함수가 시작되고,
12 // main 함수가 종료되면 C++ 프로그램이 종료된다.
13 int main() {
14     // 출력은 cout을 통해 할 수 있다.
15     // cout 뒤에 <<(출력 스트림 연산자)를 통해 출력하고자 하는 값을 작성해주자.
16     cout << "Hello inti!" << endl;
17
18     // 함수를 종료하는 return 문, 이 줄이 실행이 되면 main 함수가 종료된다.
19     return 0;
20 }
```

cout이란?

cout은 character out의 약자로 말 그대로 문자(character)를 출력(out)하는 역할을 하는 객체이다.
cout 뒤에 <<(출력 스트림 연산자)를 붙여 출력하고자 하는 값을 붙여줄 수 있으며,
아래처럼 문자열을 출력하고 싶다면 쌍따옴표를 붙여서 문자열을 작성한 후 <<로 붙여주면된다.

```
10 // C++ 프로그램의 시작이자 끝을 담당하는 main 함수
11 // C++ 프로그램이 시작되면 main 함수가 시작되고,
12 // main 함수가 종료되면 C++ 프로그램이 종료된다.
13 int main() {
14     // 출력은 cout을 통해 할 수 있다.
15     // cout 뒤에 <<(출력 스트림 연산자)를 통해 출력하고자 하는 값을 작성해주자.
16     cout << "Hello inti!" << endl;
17
18     // 함수를 종료하는 return 문, 이 줄이 실행이 되면 main함수가 종료된다.
19     return 0;
20 }
```

실행은 어떻게?

기본적으로 C++ 프로그램은 컴파일이라는 과정을 거쳐, 내가 짠 코드를 실행시킬 수 있다.
여기서 컴파일 단축키는 f5이다.

```
// C++ 프로그램이 시작되면 main 함수가 시작되고,  
// main 함수가 종료되면 C++ 프로그램이 종료된다.  
int main() {  
    // 출력은 cout을 통해 할 수 있다.  
    // cout 뒤에 <<(출력 스트림 연산자)  
    cout << "Hello inti!" << endl;  
  
    // 함수를 종료하는 return 문, 이 줄  
    return 0;  
}
```

Microsoft Visual Studio 디버그 콘솔

Hello inti!

C:\Users\yt430\source
었습니다(코드: 0개).
이 창을 닫으려면 아무

입력은 어떻게?

C++에서 입력은 cin을 통해 할 수 있다.

다만 지금은 변수를 배우지 않았으므로, 이렇게 있다 정도만 확인하고 다음 시간에 제대로 배워보도록 하자.

```
int main() {  
    int a = 0;  
  
    cin >> a;  
    cout << a << endl;  
    return 0;  
}
```

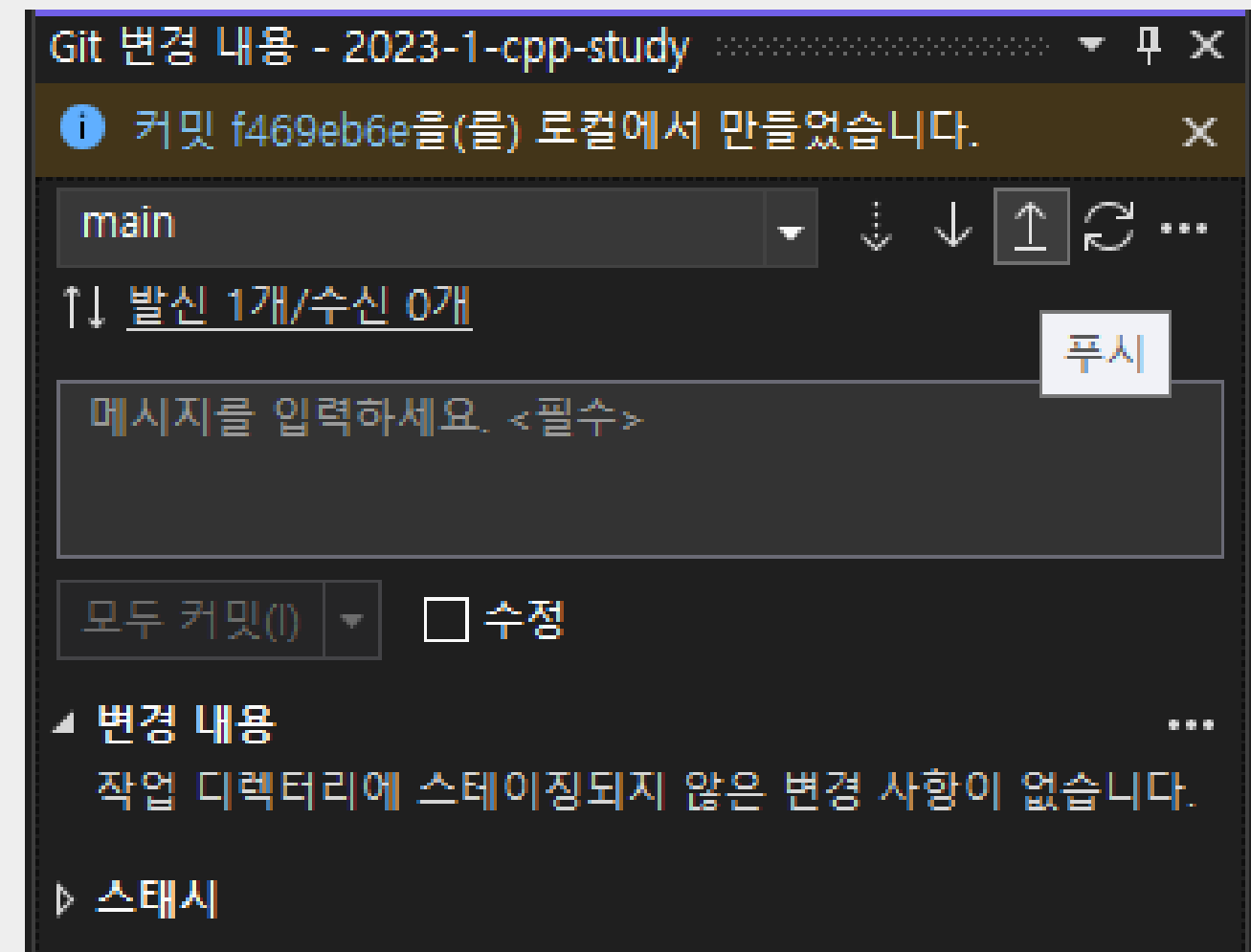
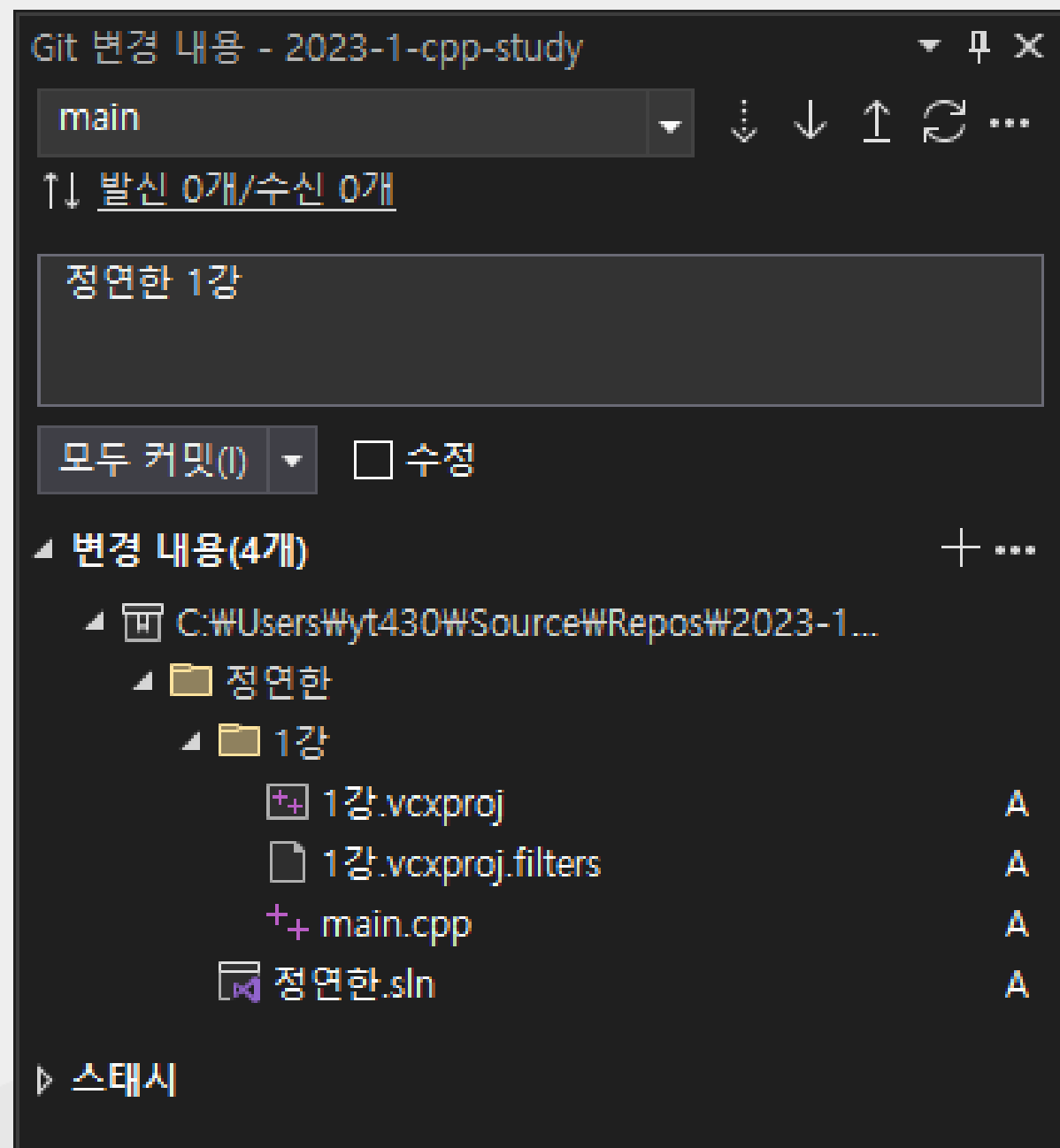
Microsoft V
10
10
C:\Users\
었습니다(
이 창을 닫

제출은 어떻게?

우리는 아까전 깃허브에서 리포를 복제해왔다.

그 뒤 복제 된 리포에서 작업을 진행했고, 이제 다시 깃허브로 작업한 파일을 올릴 차례이다.

이런식으로 커밋 메시지를 작성해주고, 모두 커밋을 눌러 준 후, 우측 상단에 푸시 또는 동기화 버튼을 누르면 끝!



제출은 어떻게?

만약 푸시 버튼을 눌렀는데 아래 화면이 뜰 경우 당황하지 말고 Sign in with your browser를 눌러 로그인하고 다시 푸시 또는 동기화를 시도해주자!

