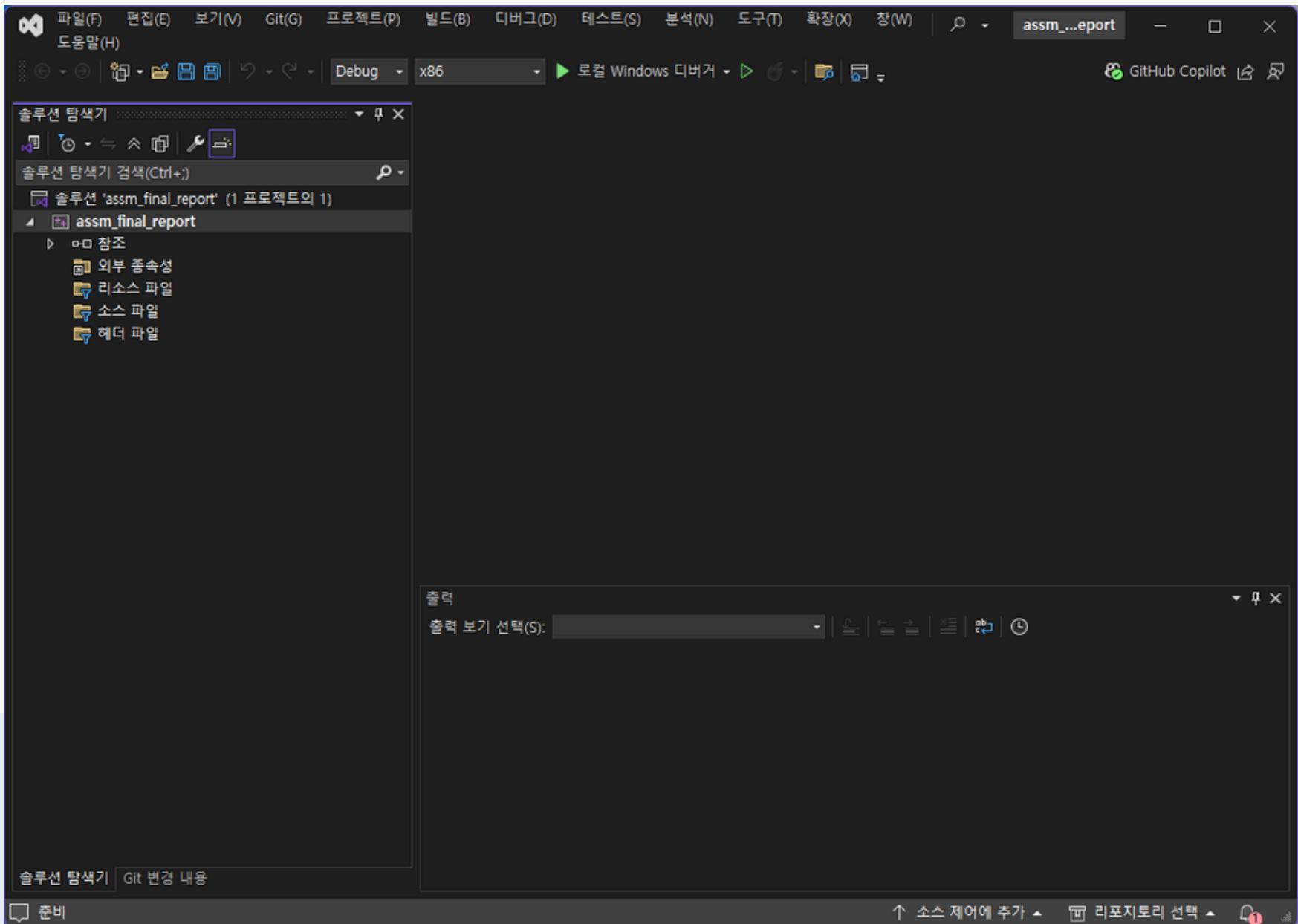


ASSEMBLY PROGRAMMING

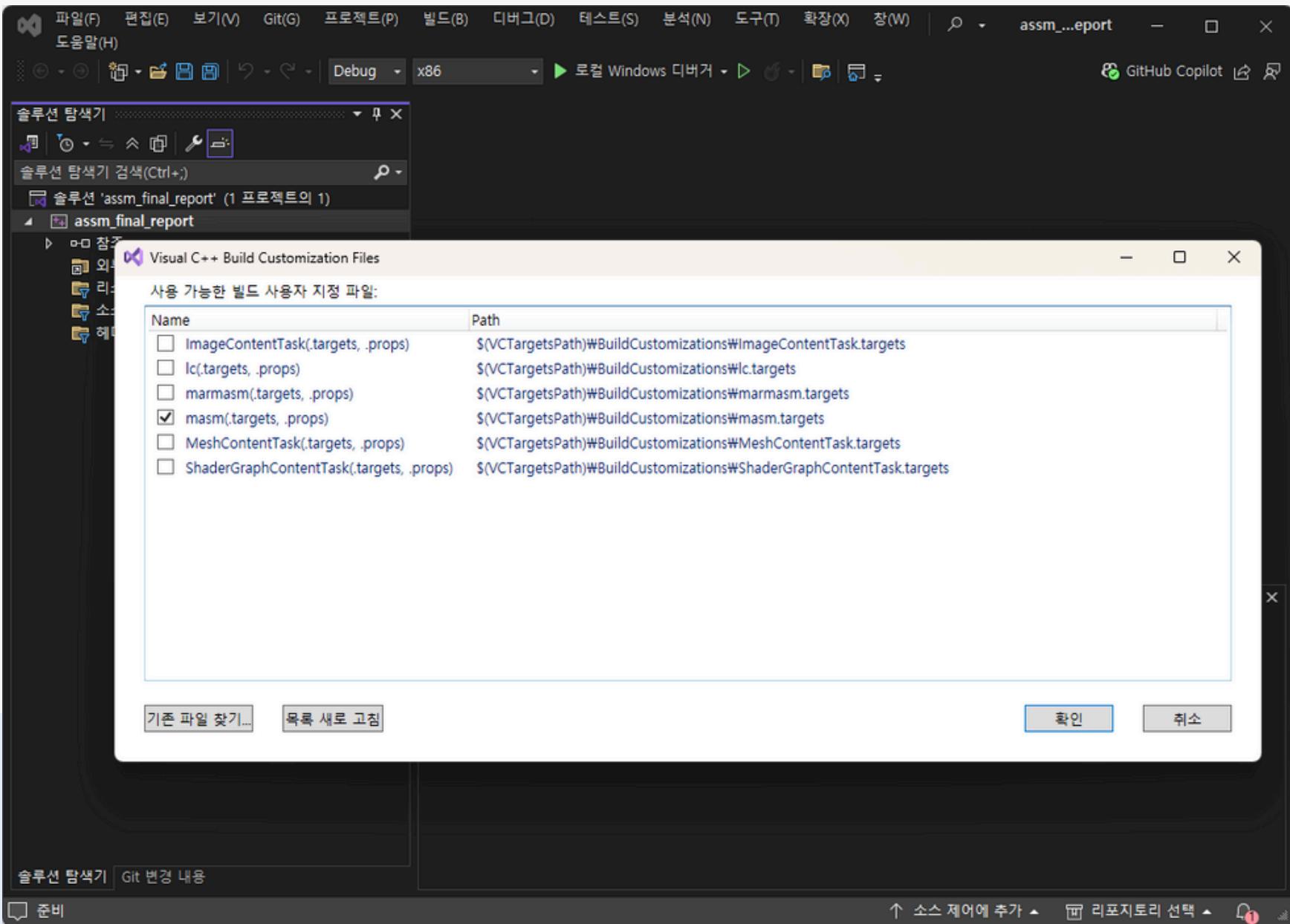
BUBBLE SORT를  
**C + ASM, LIBRARY, DLL**

2021752038 조현창

## 새 프로젝트 생성



## MASM 설정



어셈블리어를 인식하기 위해 MASM 설정을 한다

프로젝트 우클릭 → 빌드 종속성 → 사용자 지정 빌드 → masm(.targets, .props) 체크 → 확인

## BUBBLE SORT 어셈블리 코드 작성

The screenshot shows the assembly code for the `bubbleSortASM` procedure. The code implements a bubble sort algorithm using registers ECX, ECX, EDX, and EDI. It includes labels for the outer loop, inner loop, and swap conditions. The assembly code is as follows:

```

1 .686
2 .model flat, c
3 .code
4
5 bubbleSortASM PROC
6     push ebp
7     mov ebp, esp
8
9     mov esi, [ebp + 8]
10    mov ecx, [ebp + 12]
11
12    dec ecx
13    cmp ecx, 0
14    jle END_SORT
15
16    OUTER_LOOP:
17    push ecx
18    mov edx, ecx
19    mov edi, esi
20
21    INNER_LOOP:
22    mov eax, [edi]
23    mov ebx, [edi + 4]
24
25    cmp eax, ebx
26    jle NO_SWAP
27
28    mov [edi], ebx
29    mov [edi + 4], eax
30
31    NO_SWAP:
32    add edi, 4
33    dec edx
34    jnz INNER_LOOP
35
36    pop ecx
37    dec ecx
38    jnz OUTER_LOOP
39
40    END_SORT:
41    mov esp, ebp
42    pop ebp
43    ret
44
45 bubbleSortASM ENDP
46 END

```

## MAIN역할을 할 C 파일과, 헤더 파일 생성

The screenshot shows the `main.c` and `BubbleSort.h` files. The `main.c` file contains the main function which prints the initial array and calls the `bubbleSort` function. The `BubbleSort.h` file contains the declaration of the `bubbleSort` function. The code is as follows:

```

main.c
1 #include <stdio.h>
2
3 extern void bubbleSort(int* arr, int n);
4
5 int main() {
6     int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
7     int n = sizeof(arr) / sizeof(arr[0]);
8     int i;
9
10    printf("정렬 전: ");
11    for (i = 0; i < n; i++) printf("%d ", arr[i]);
12    printf("\n");
13
14    bubbleSort(arr, n);
15
16    printf("정렬 후: ");
17    for (i = 0; i < n; i++) printf("%d ", arr[i]);
18    printf("\n");
19
20    return 0;
21 }

```

```

BubbleSort.h
1 #pragma once
2
3 #ifndef BUBBLESORT_H
4 #define BUBBLESORT_H
5
6 extern void bubbleSort(int* arr, int n);
7
8#endif

```

사진에 있는 MAIN.C 파일은 자기 파일 내용만 알기 때문에 어셈블리로 작성된 BUBBLE SORT 함수가 호출되면 에러가 발생합니다. 이때 헤더파일이 다른 파일에 정의 되어있다고 알려주는 역할을 해줍니다.

즉, 다른 소스 파일에 작성된 함수의 존재나 사용법을 컴파일러에게 미리 알려주어 문제없이 링크하기 위해서 사용됩니다.

## 실행

The screenshot shows the Microsoft Visual Studio interface during the execution of a program named 'assm\_final\_report'. The main window displays two files: 'BubbleSort.asm' and 'main.c'. The assembly code in 'BubbleSort.asm' contains instructions to sort an array. The C code in 'main.c' includes a call to this assembly function. The output window shows the results of the execution:

```
정렬 전: 64 34 25 12 22 11 90
정렬 후: 11 12 22 25 34 64 90

C:\Users\USER\source/repos\assm_final_report\Debug\assm_final_report.exe(프로세스 27524)이(가) 0 코드(0x0)와 함께 종료되었습니다.

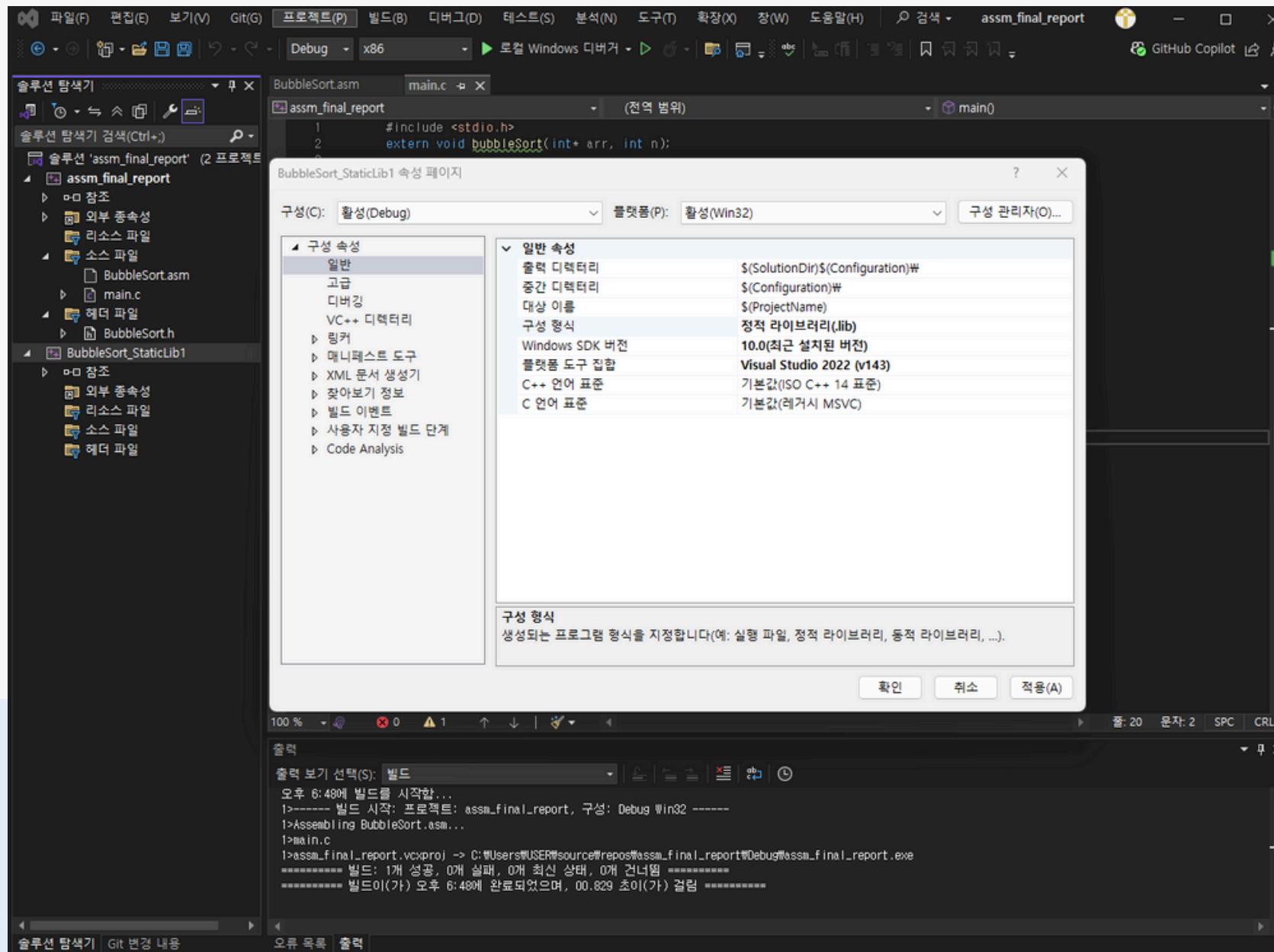
이 창을 닫으려면 아무 키나 누르세요...
```

The build log at the bottom of the output window indicates a successful build:

```
1>main.c
1>assm_final_report.vcxproj -> C:\Users\USER\source\repos\assm_final_report\Debug\assm_final_report.exe
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 =====
===== 빌드이(가) 오후 6:48에 완료되었으며, 00.829 초이(가) 걸림 =====
```

실행을 하게 되면 어셈블리어로 작성된 BubbleSort 함수가 Main인 C 파일에서 정상적으로 호출되고 작동하는 모습을 볼 수 있습니다

## 솔루션에 프로젝트 추가 후 구성형식을 정적 라이브러리로 변경



## 앞에 작성했던 어셈블리 소스파일과 헤더파일 복사 해오기

```

;include <stdio.h>
extern void bubbleSort(int* arr, int n);

.b64
.model flat, c
.code

bubbleSort PROC
    push ebp
    mov ebp, esp
    mov esi, [ebp + 8]
    mov ecx, [ebp + 12]
    dec ecx
    cmp ecx, 0
    jle END_SORT
    OUTER_LOOP:
    push ecx
    mov edx, ecx
    mov edi, esi
    INNER_LOOP:
    mov eax, [edi]
    mov ebx, [edi + 4]
    cmp eax, ebx
    jle NO_SWAP
    mov [edi], ebx
    mov [edi + 4], eax
    NO_SWAP:
    add edi, 4
    dec edx
    jnz INNER_LOOP
    pop ecx
    dec ecx
    END_SORT:
    mov esp, ebp
    pop ebp
    ret
bubbleSort ENDP

main()
{
    int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int i;

    printf("정렬 전: ");
    for (i = 0; i < n; i++) printf("%d ", arr[i]);
    printf("\n");

    bubbleSort(arr, n);

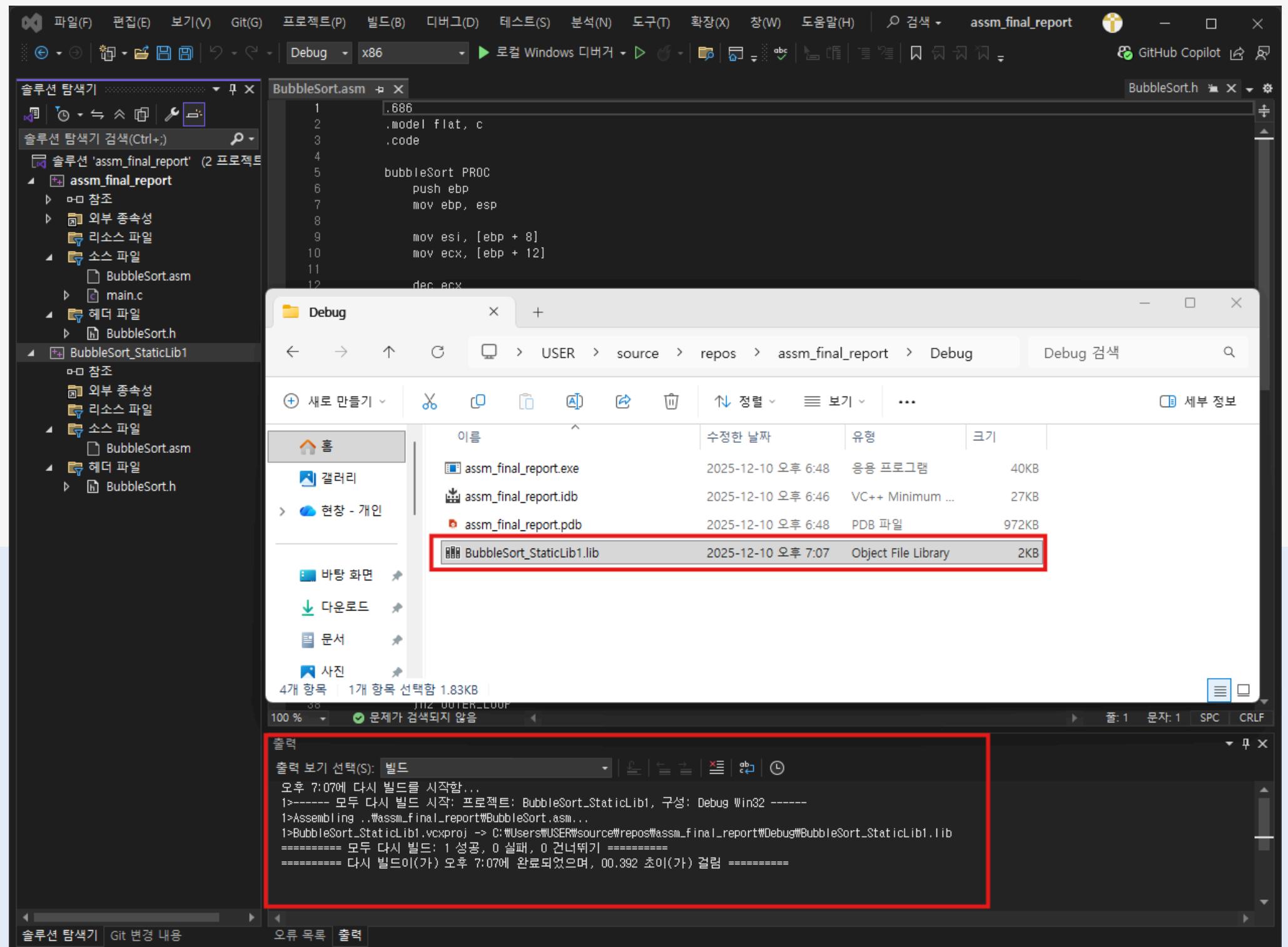
    printf("정렬 후: ");
    for (i = 0; i < n; i++) printf("%d ", arr[i]);
    printf("\n");
}

return 0;

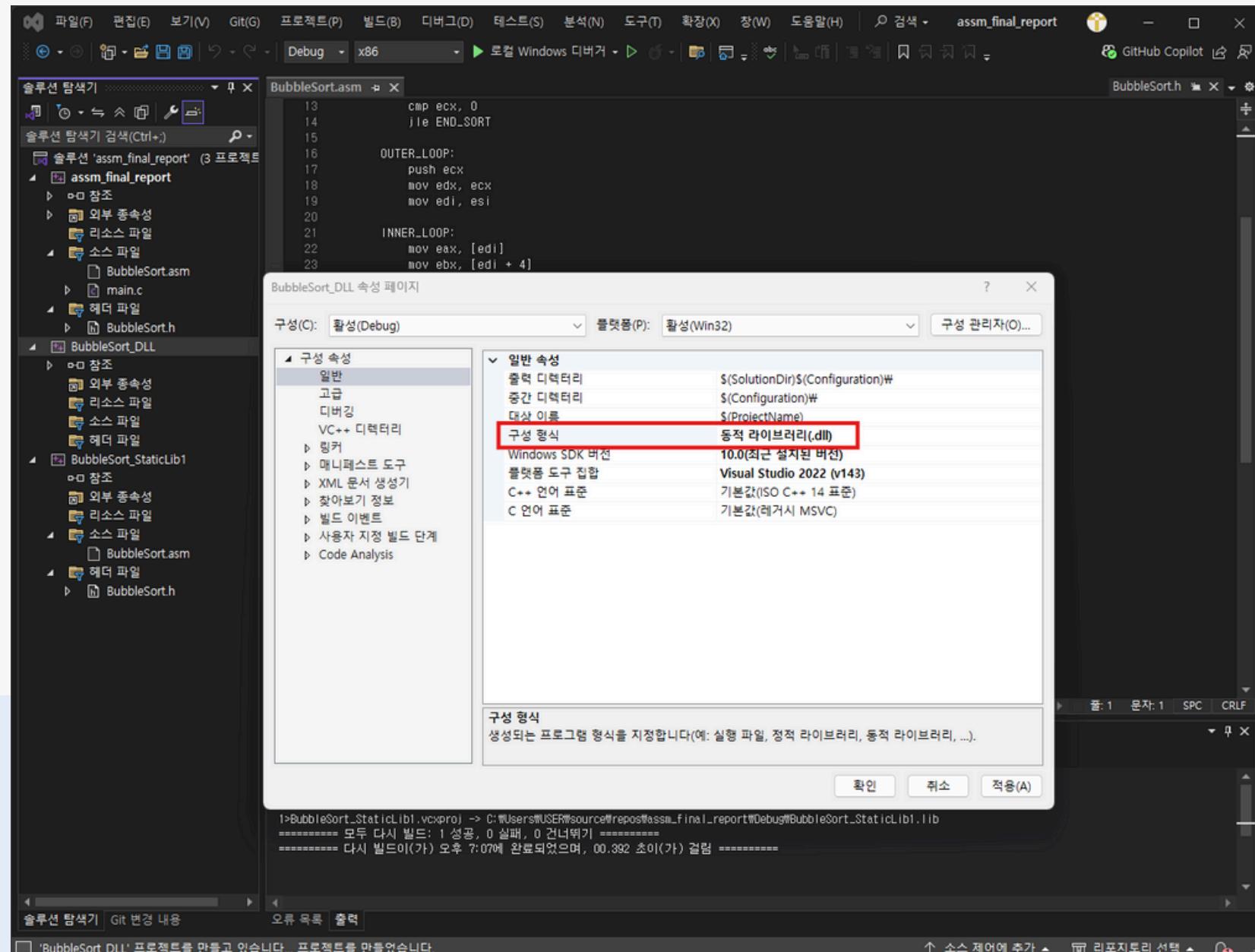
```

- 솔루션 우클릭 → 추가 → 새 프로젝트 → 빙 프로젝트
- 프로젝트 우클릭 → 속성 → 구성형식 → 정적 라이브러리 선택

## 빌드 및 실행

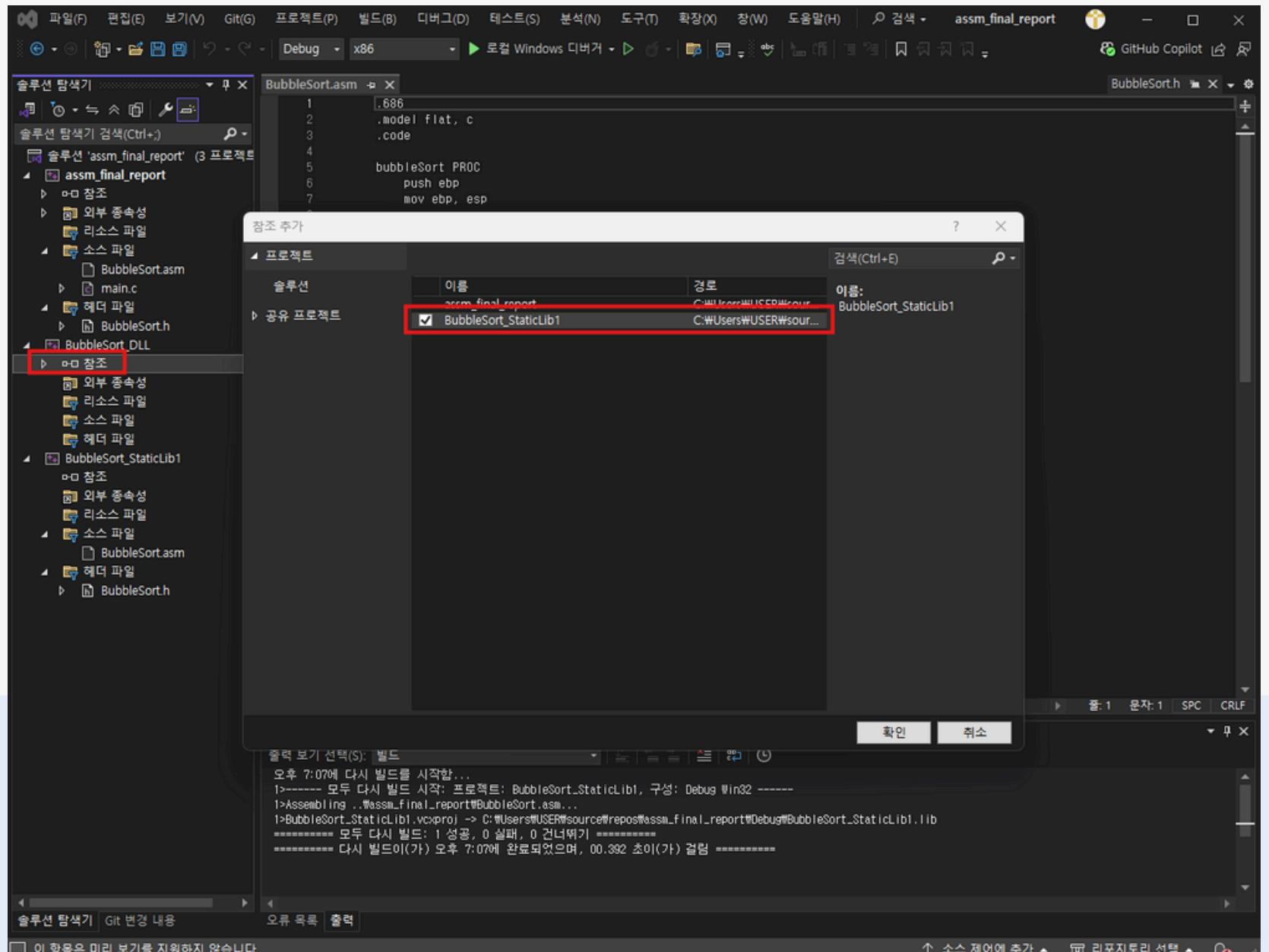


## 솔루션에 프로젝트 추가 후 구성형식을 동적 라이브러리로 변경



- 솔루션 우클릭 → 추가 → 새 프로젝트 → 빙 프로젝트
- 프로젝트 우클릭 → 속성 → 구성형식 → 동적 라이브러리 선택

## 정적 라이브러리(Lib) 연결



- 참조 우클릭 → 참조 추가 → 만들어둔 정정 라이브러리 선택 → 확인

DLL이 만들어질 때 만들어둔 어셈블리 부품(Lib)를 가져다 쓰기 위함입니다.

## DLL 기본 설정과 함수 이름 외부 공개(EXPORT)를 위한 내보내기 설정

The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the Solution Explorer displays the project structure for 'BubbleSort\_DLL'. It includes files like 'asm\_final\_report', 'BubbleSort.asm', 'main.c', 'BubbleSort.h', 'BubbleSort\_StaticLib1', and 'BubbleSort\_DLLdef'. The 'BubbleSort\_DLLdef' file is selected and contains the following code:

```

LIBRARY SortDLL
EXPORTS
    bubbleSort

```

The 'asm\_final\_report' file contains the 'DllMain' function implementation. The 'Output' window at the bottom shows the build log:

```

오후 7:34에 빌드를 시작합니다...
1>----- 빌드 시작: 프로젝트: BubbleSort_DLL, 구성: Debug Win32 -----
1>main.c
1>BubbleSort_DLL.vcxproj -> C:\Users\USER\source\repos\asm_final_report\Debug\BubbleSort_DLL.dll
===== 빌드: 1개 성공, 0개 실패, 1개 최신 상태, 0개 건너뜀 =====
===== 빌드(가) 오후 7:34에 완료되었습니다, 00.566 초이(가) 걸립 =====

```

## 빌드 및 실행 확인

The screenshot shows the Microsoft Visual Studio interface after a successful build. The 'Solution Explorer' shows the 'Debug' folder containing the generated files:

- assm\_final\_report.exe
- assm\_final\_report.idb
- assm\_final\_report.pdb
- BubbleSort\_DLL.dll** (highlighted with a red box)
- BubbleSort\_DLL.pdb
- BubbleSort\_StaticLib1.lib

The 'Output' window at the bottom shows the build log again:

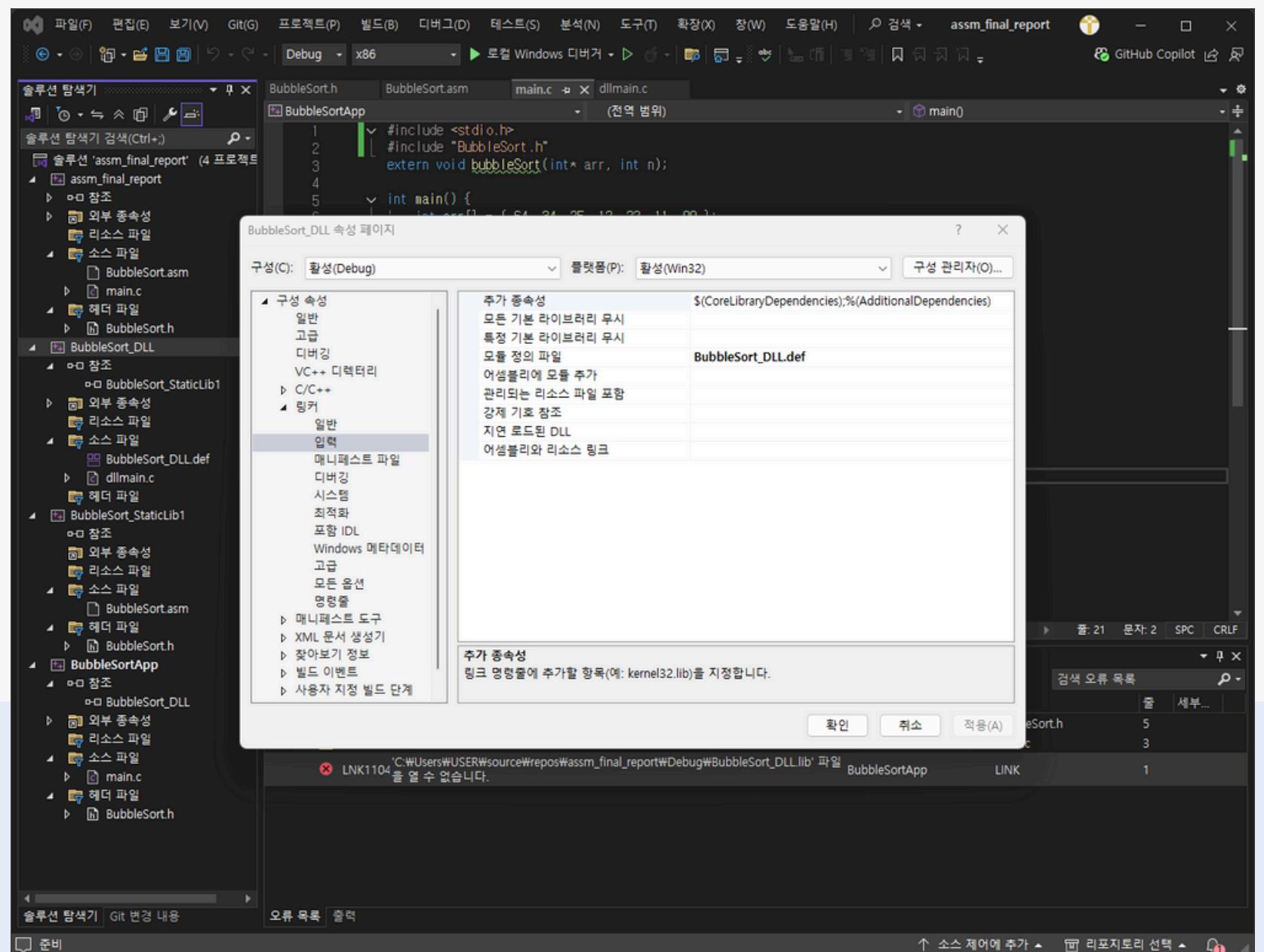
```

오후 7:34에 빌드를 시작합니다...
1>----- 빌드 시작: 프로젝트: BubbleSort_DLL, 구성: Debug Win32 -----
1>main.c
1>BubbleSort_DLL.vcxproj -> C:\Users\USER\source\repos\asm_final_report\Debug\BubbleSort_DLL.dll
===== 빌드: 1개 성공, 0개 실패, 1개 최신 상태, 0개 건너뜀 =====
===== 빌드(가) 오후 7:34에 완료되었습니다, 00.566 초이(가) 걸립 =====

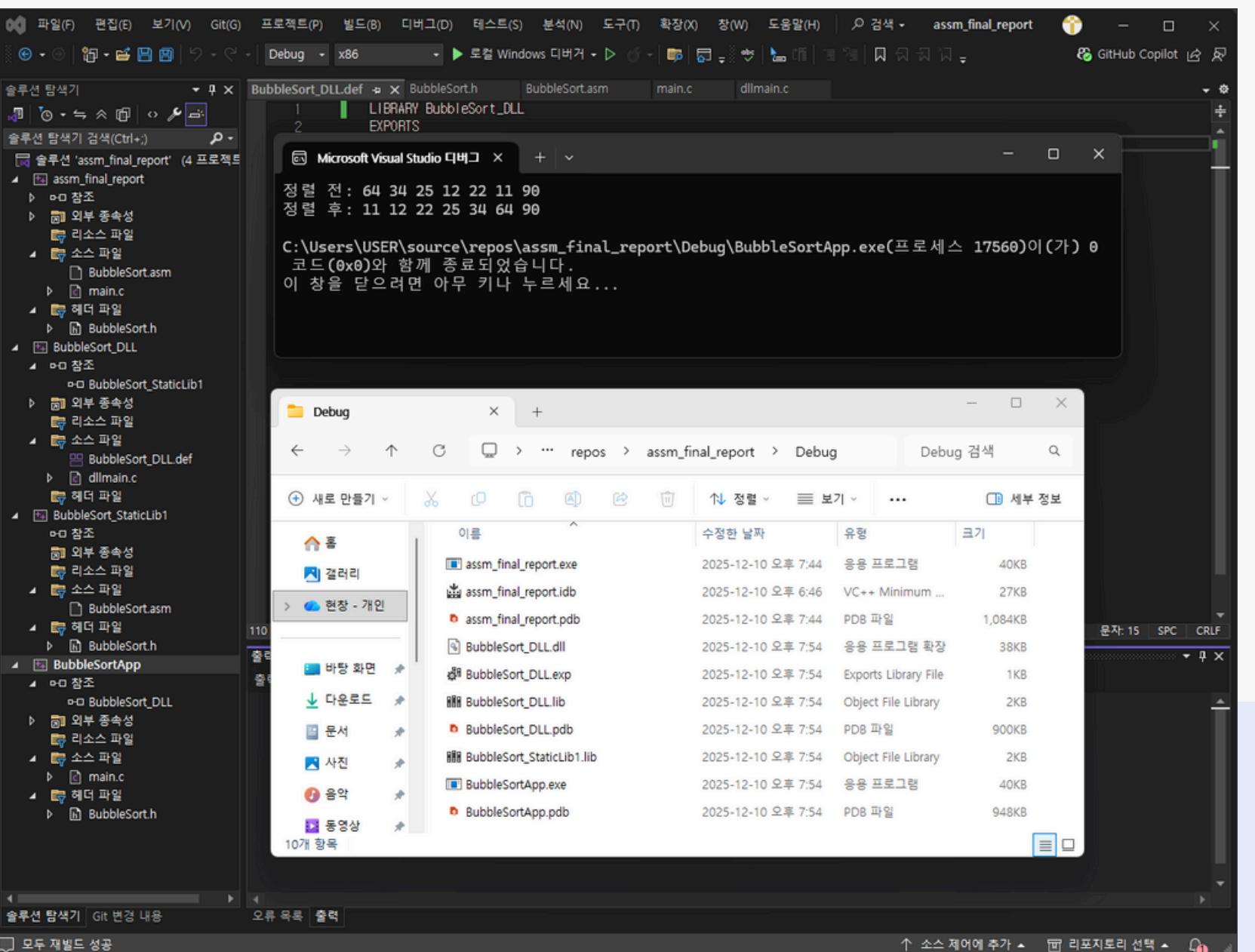
```

- DLL의 필수 구성 요소 파일 생성
- 소스파일 우클릭 → 추가 → 새항목 → 텍스트 파일 → 이름.def → 확인
  - 외부 프로그램들이 확인을 하고 사용 할 수 있도록 하는 설정

## 모듈 정의 파일 설정



## 빌드 및 실행 확인



# 점수: 10점

- 제 시간에 완료
- 모든 요구사항 문제 없이 완료



감사합니다

레퍼런스: <https://gemini.google.com/>