

어셈블리어 과제

PRESENTATION

2022564031
이상헌



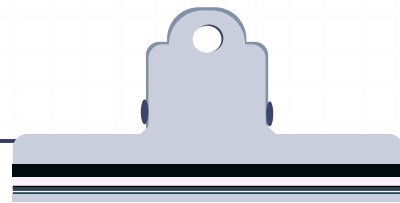


목차

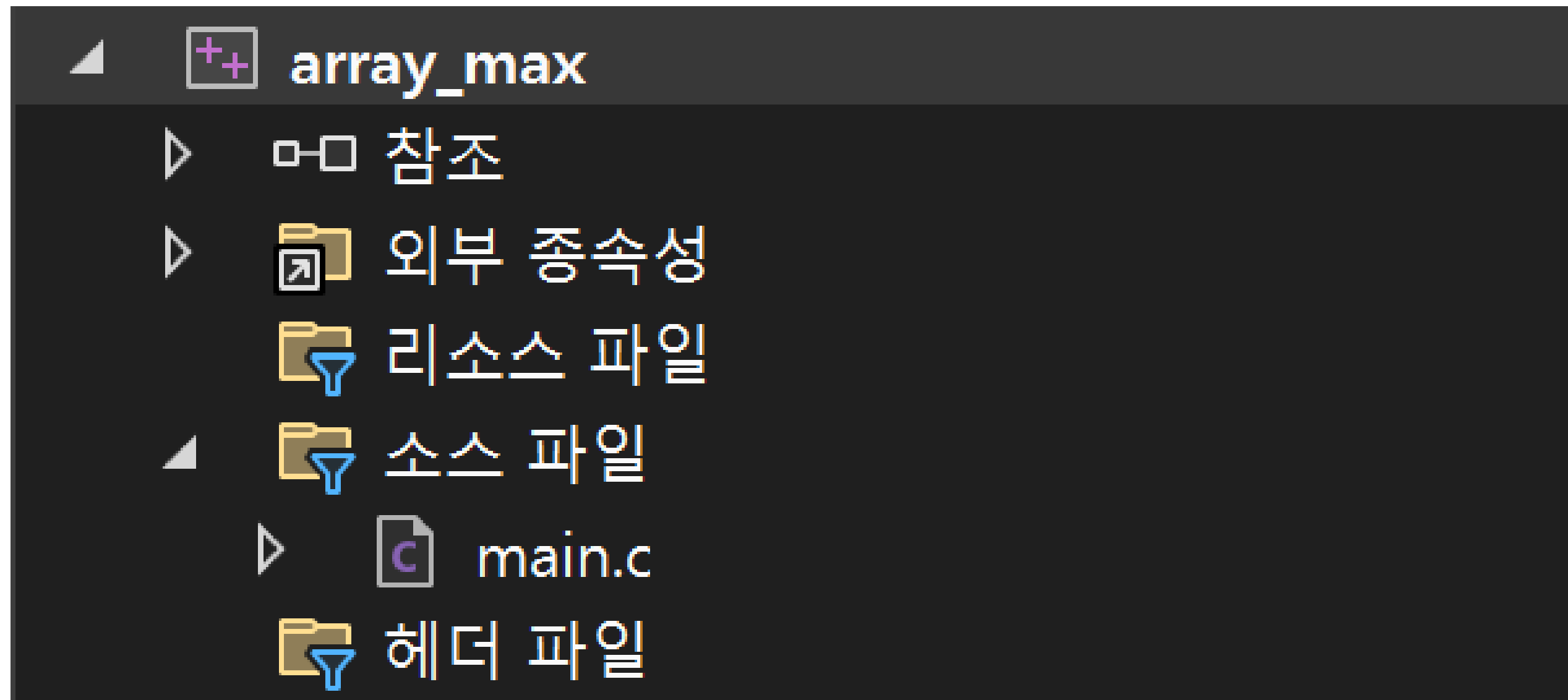
01 asm+c

02 lib

03 dll



01 asm+c



프로젝트(array_max)를 만들어 소스 파일로 main.c파일을 만들어줍니다.

```

main.c  X
array_max  (전역 범위)
1  #include <stdio.h>
2
3  int find_max(int arr[], int len)
4  {
5      int max_value = 0;
6
7      __asm {
8          mov ecx, len; len을 ECX에 저장
9          cmp ecx, 0
10         jle no_elements; len <= 0 이면 종료
11
12         mov esi, arr; 배열 주소 로드
13         mov eax, [esi]; 첫 번째 값
14         mov max_value, eax; max = arr[0]
15         add esi, 4; 다음 요소로 이동
16         dec ecx; 요소 하나 사용했으므로 감소
17
18         loop_start :
19         cmp ecx, 0
20         je end_loop; 배열 끝났으면 종료
21
22         mov eax, [esi]; 현재 arr[i]
23         cmp eax, max_value
24         jle skip_update; max보다 작으면 건너뛰기
25
26         mov max_value, eax; max 갱신
27
28         skip_update :
29         add esi, 4; 다음 요소
30         dec ecx

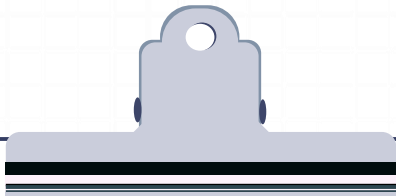
```

```

main.c  X
array_max  (전역 범위)
29         add esi, 4; 다음 요소
30         dec ecx
31         jmp loop_start
32
33         end_loop :
34         jmp done
35
36         no_elements :
37         mov max_value, 0; 요소 없으면 0 반환
38
39         done :
40     }
41
42     return max_value;
43 }
44
45 int main()
46 {
47     int arr[5];
48
49     printf("정수 5개를 입력하세요:\n");
50     for (int i = 0; i < 5; i++)
51         scanf_s("%d", &arr[i]);
52
53     int max_val = find_max(arr, 5);
54
55     printf("가장 큰 값: %d\n", max_val);
56
57     return 0;
58 }

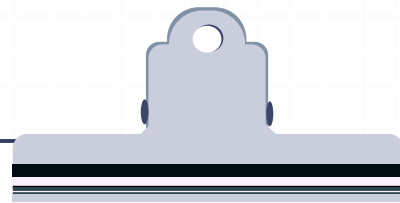
```

코드는 입력받은 배열 내에서 최대값을 찾는 코드입니다. __asm을 활용하여 어셈블리 코드를 작성하였습니다.

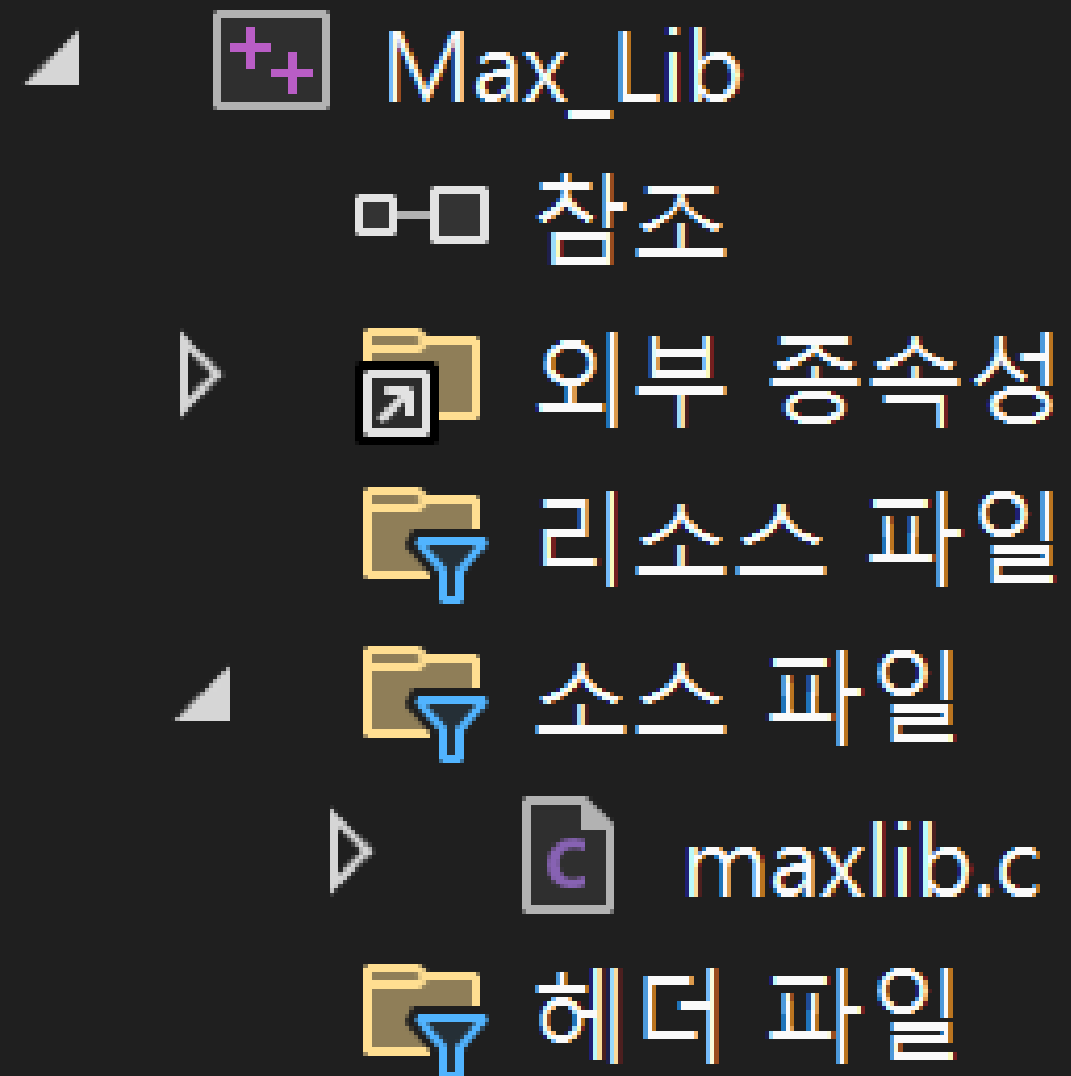


```
Microsoft Visual Studio 디버그 × + ▾ − □ ×  
정수 5개를 입력하세요 :  
1  
3  
19  
4  
2  
가장 큰 값 : 19  
  
C:\Assem\Project32_VS2022\Debug\array_max.exe(프로  
세스 8240)이(가) 0 코드(0x0)와 함께 종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요...|
```

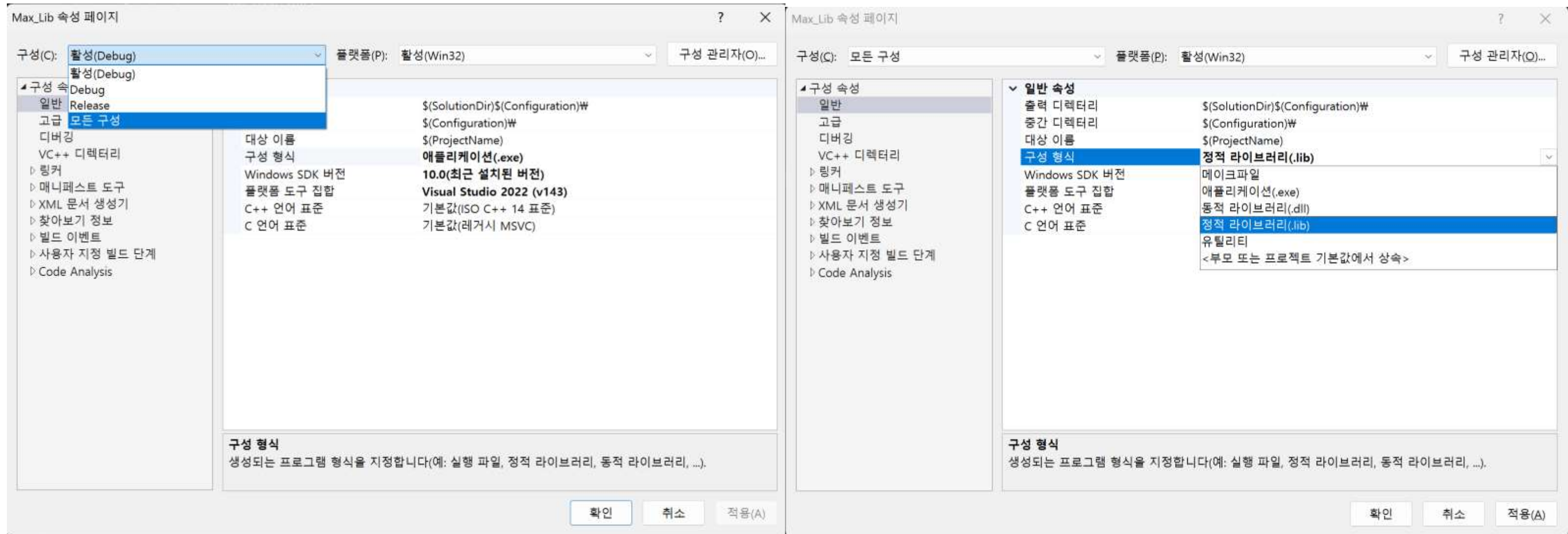
정상적으로 작동하는 모습을 볼 수 있습니다.



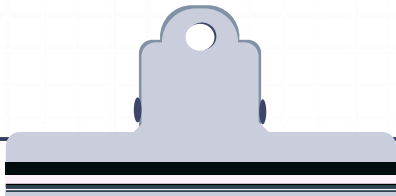
02 lib



새로운 프로젝트(Max_Lib)를 만들고 소스 파일로 maxlib.c 파일을 만들어주었습니다.

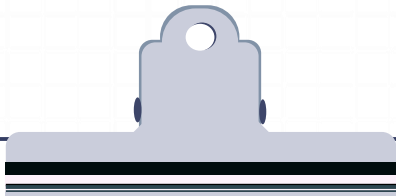


새로운 프로젝트(Max_lib)에서 우클릭을 하여 속성에 들어가줍니다.
구성: 모든 구성 구성 형식: 정적 라이브러리(lib)를 선택하고 적용해줍니다.



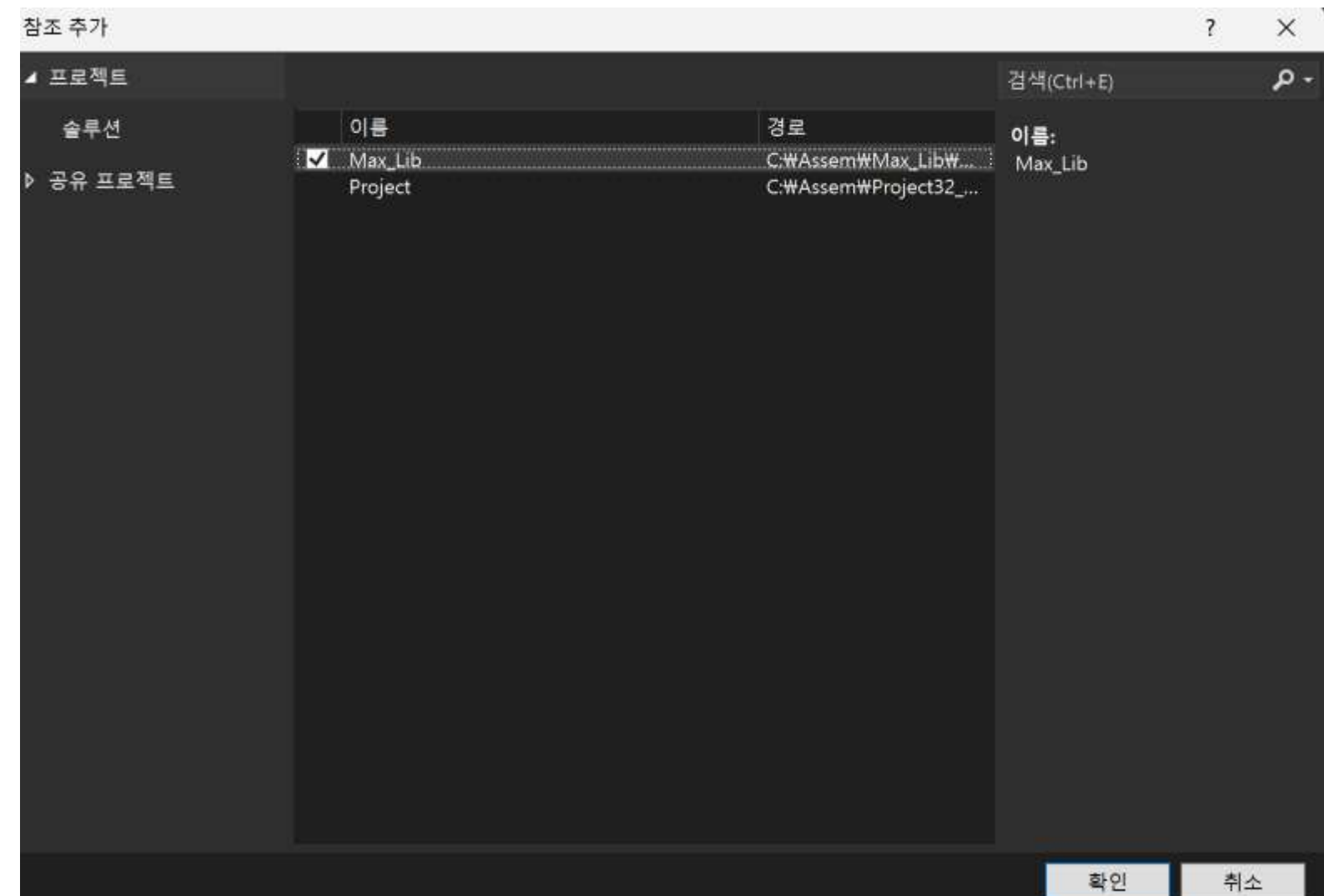
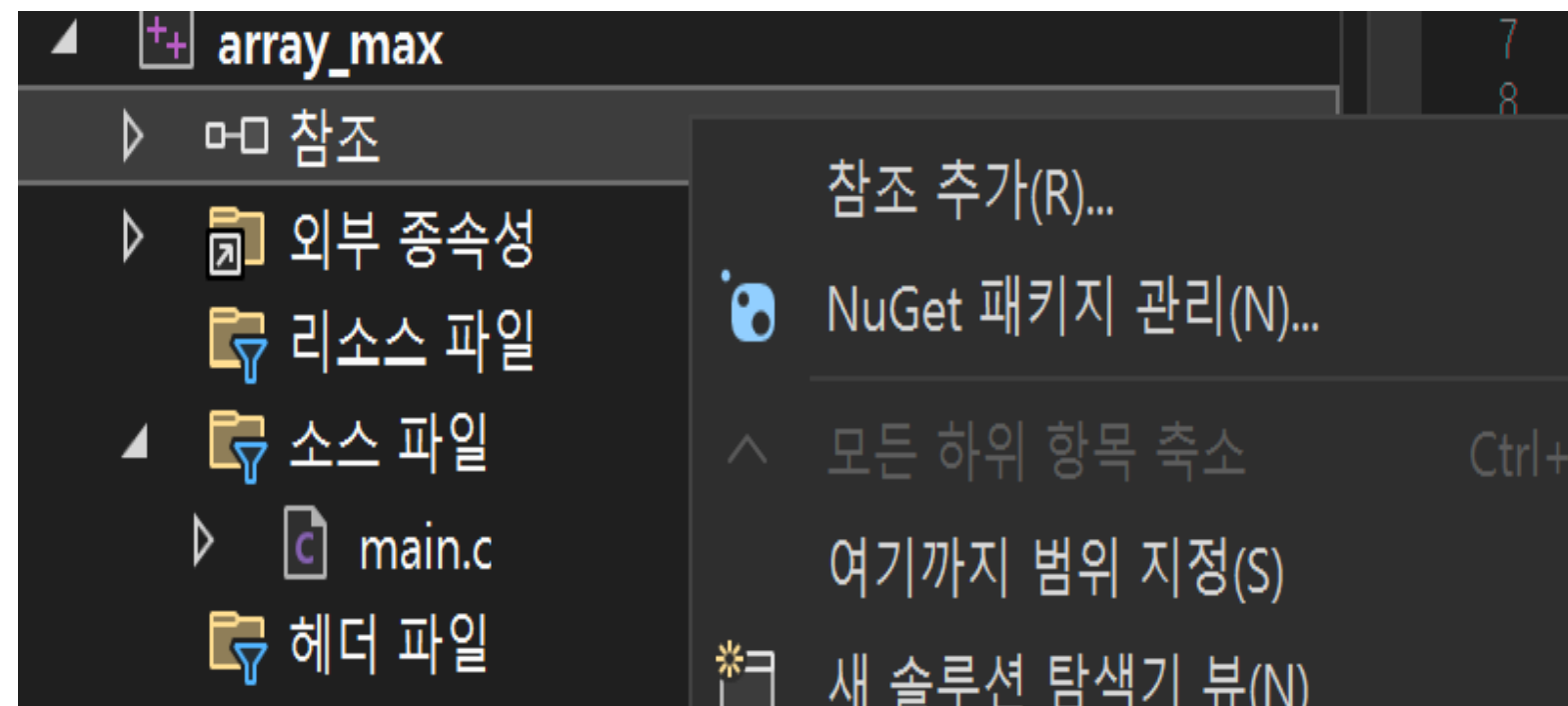
```
1  #include <stdio.h>
2
3  int find_max(int arr[], int len)
4  {
5      int max_value = 0;
6
7      __asm {
8          mov ecx, len; len을 ECX에 저장
9          cmp ecx, 0
10         jle no_elements; len <= 0 이면 종료
11
12         mov esi, arr; 배열 주소 로드
13         mov eax, [esi]; 첫 번째 값
14         mov max_value, eax; max = arr[0]
15         add esi, 4; 다음 요소로 이동
16         dec ecx; 요소 하나 사용했으므로 감소
17         loop_start:
18         cmp ecx, 0
19         je end_loop; 배열 끝났으면 종료
20
21         mov eax, [esi]; 현재 arr[i]
22         cmp eax, max_value
23         jle skip_update; max보다 작으면 건너뛰기
24
25         mov max_value, eax; max 갱신
26
27         skip_update:
28         add esi, 4; 다음 요소
29         dec ecx
30         jmp loop_start
31         end_loop:
32         jmp done
33         no_elements:
34         mov max_value, 0; 요소 없으면 0 반환
35         done:
36     }
37     return max_value;
38 }
```

maxlib.c파일에 기존의 main.c의 코드에서 main()을 제외한 코드들을 작성해줍니다.

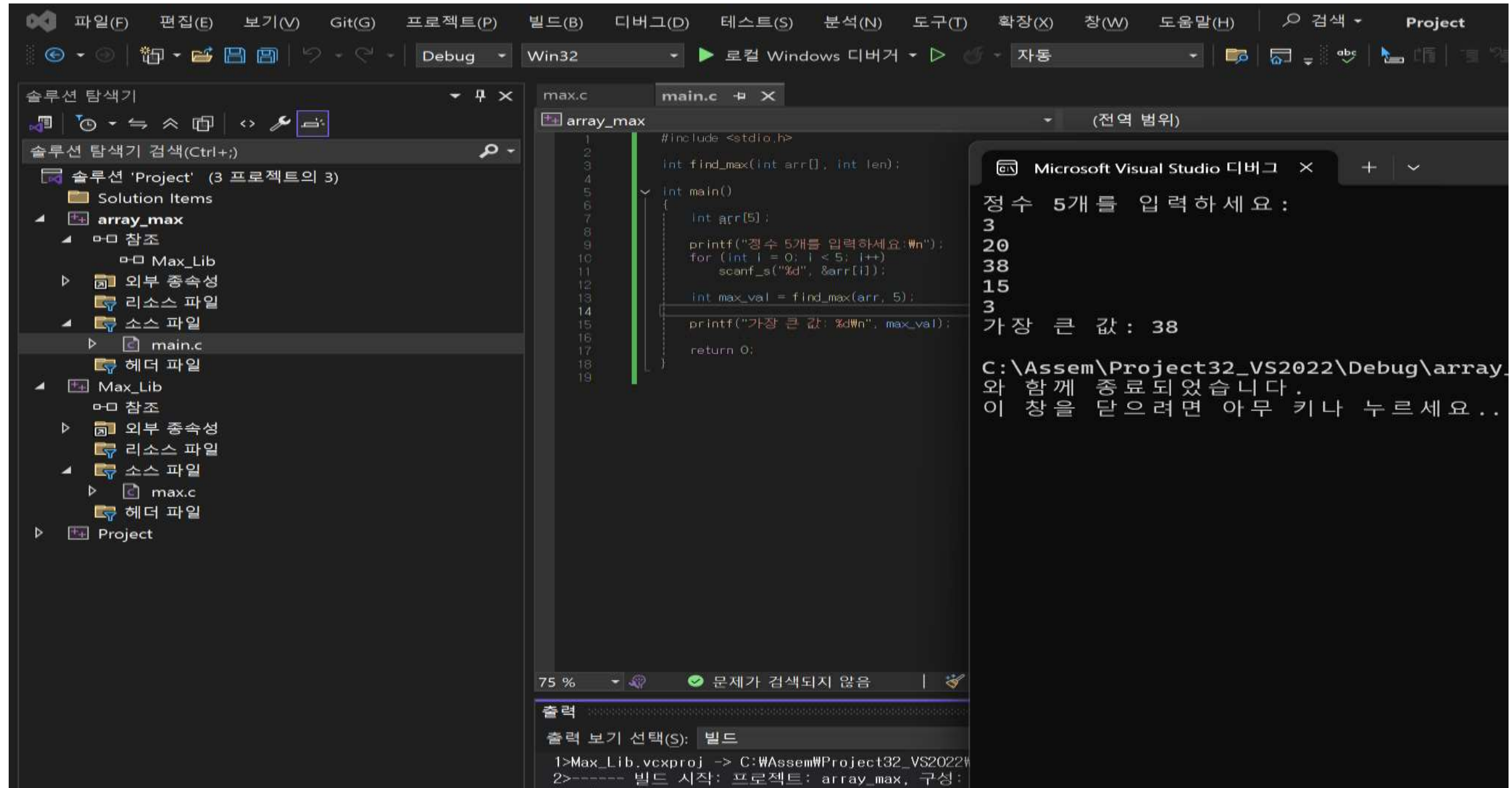


```
max.c*  main.c  X
array_max  (전
1      #include <stdio.h>
2
3      int find_max(int arr[], int len);
4
5      int main()
6      {
7          int arr[5];
8
9          printf("정수 5개를 입력하세요:\n");
10         for (int i = 0; i < 5; i++)
11             scanf_s("%d", &arr[i]);
12
13         int max_val = find_max(arr, 5);
14
15         printf("가장 큰 값: %d\n", max_val);
16
17         return 0;
18     }
```

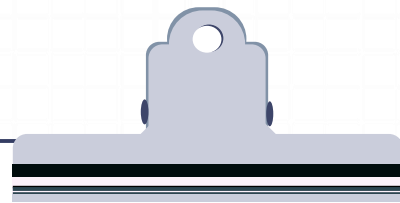
기존에 있던 함수는 라이브러리에 있으니 실행할 main.c의 코드도 바꿔줍니다.



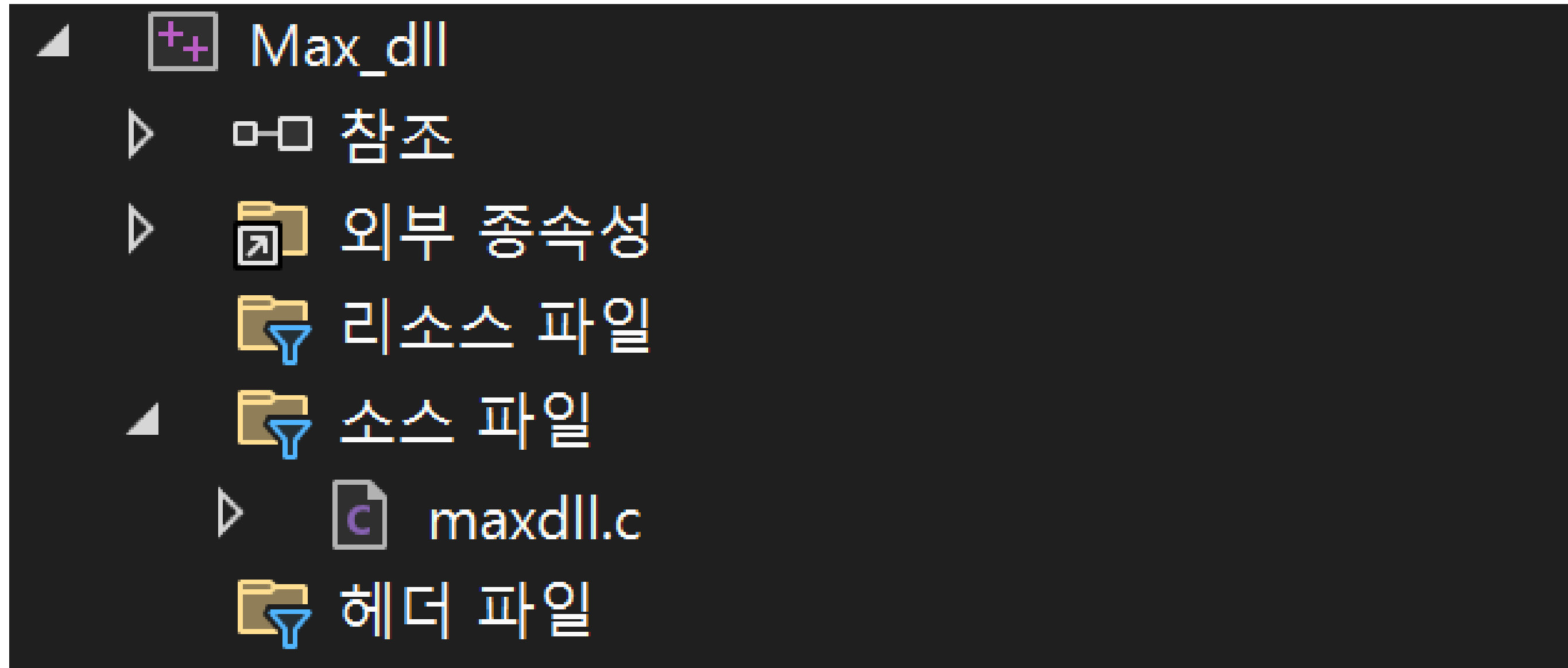
실행할 프로젝트인 array_max의 참조로 앞서 만든 라이브러리 프로젝트를 선택해줍니다.



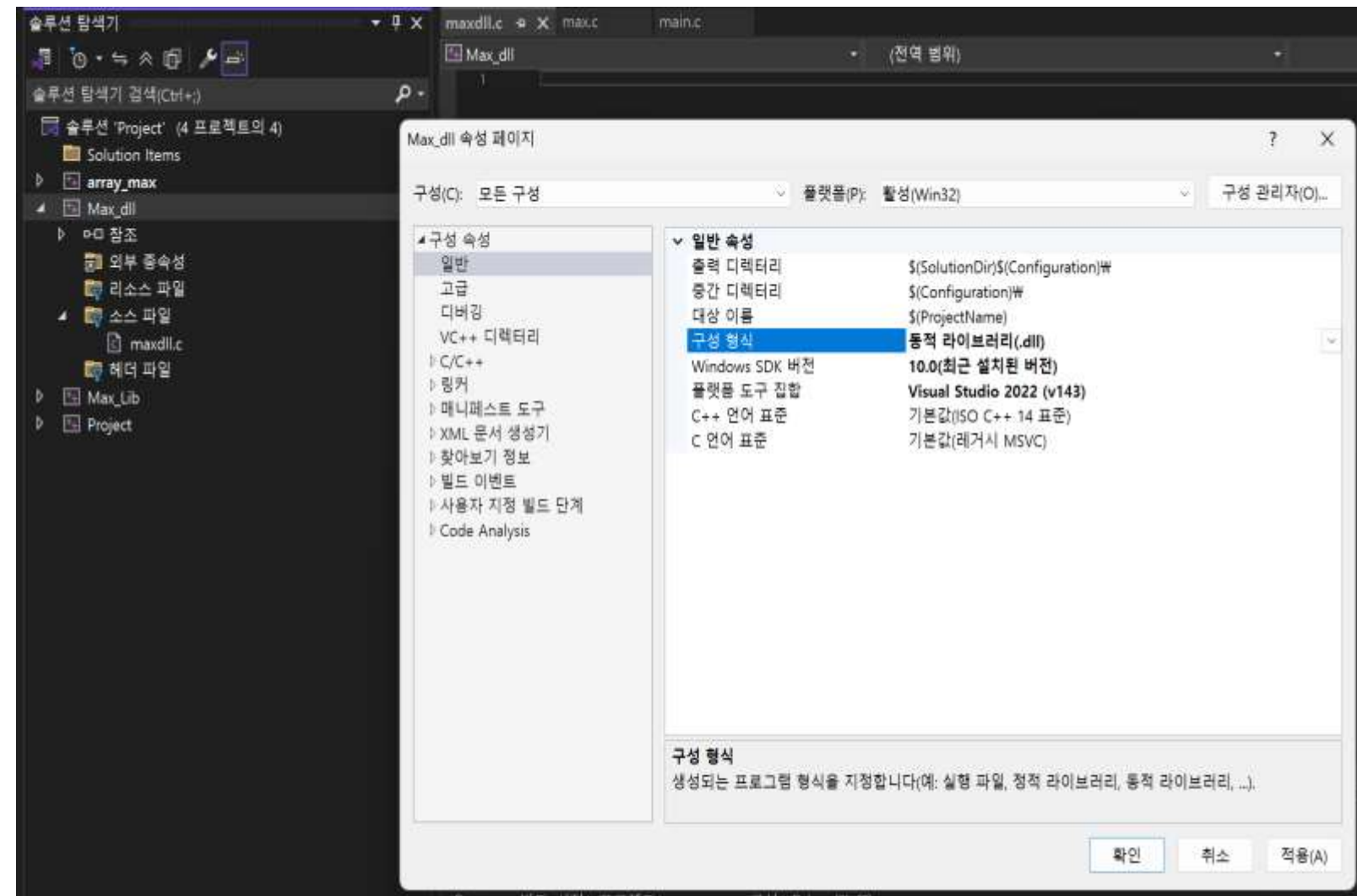
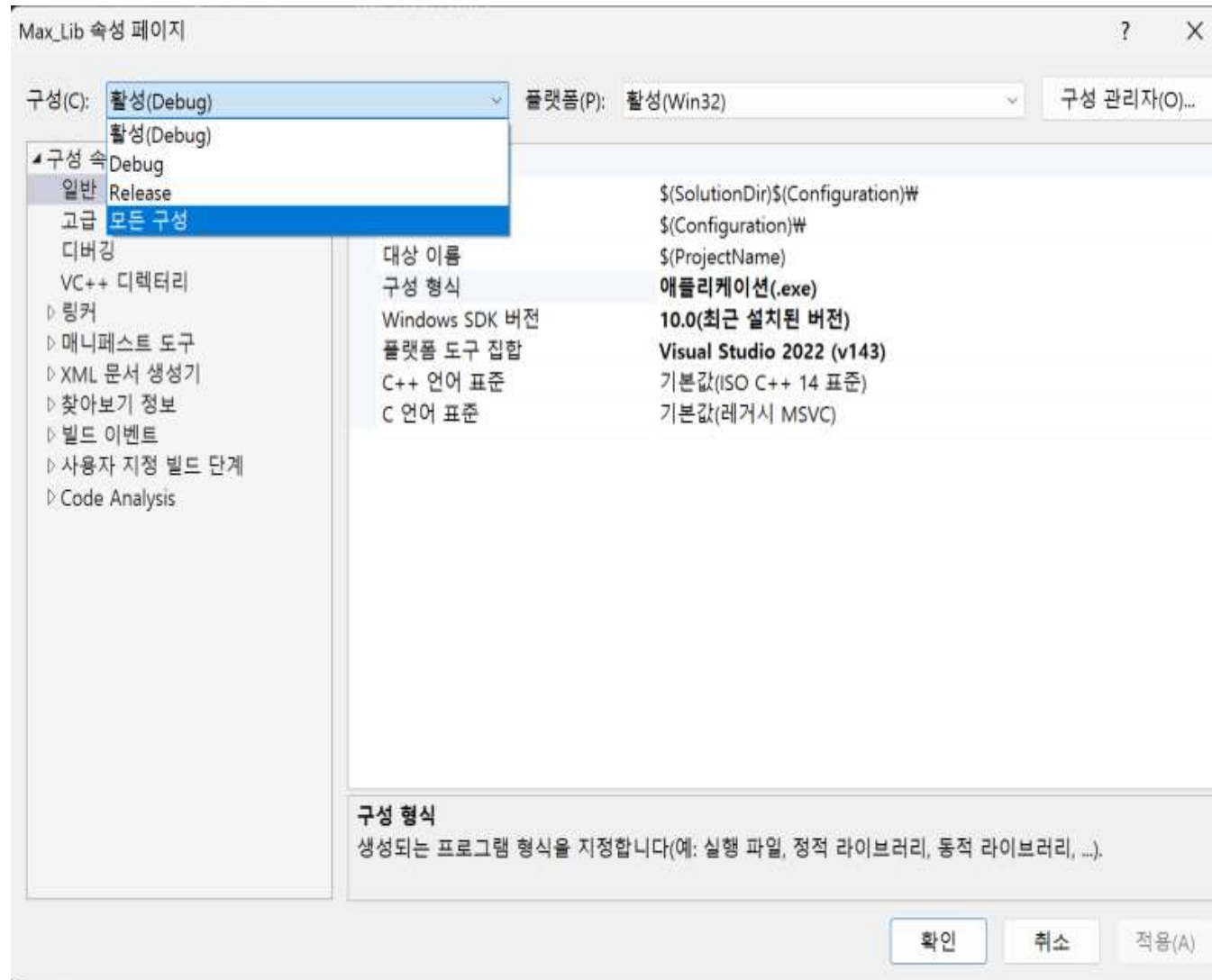
정상적으로 실행되는 것을 확인할 수 있습니다.



03 dll



이번에도 새로운 프로젝트(Max_dll)와 소스 파일 maxdll.c를 생성해줍니다.



새로운 프로젝트(Max_dll)에서 우클릭을 하여 속성에 들어가줍니다.
구성: 모든 구성 구성 형식: 동적 라이브러리(dll)를 선택하고 적용해줍니다.


```

1  #include <stdio.h>
2
3  __declspec(dllexport) int find_max(int arr[], int len)
4  {
5      int max_value = 0;
6
7      __asm {
8          mov ecx, len; len을 ECX에 저장
9          cmp ecx, 0
10         jle no_elements; len <= 0 이면 종료
11
12         mov esi, arr; 배열 주소 로드
13         mov eax, [esi]; 첫 번째 값
14         mov max_value, eax; max = arr[0]
15         add esi, 4; 다음 요소로 이동
16         dec ecx; 요소 하나 사용했으므로 감소
17         loop_start :
18         cmp ecx, 0
19         je end_loop; 배열 끝났으면 종료
20
21         mov eax, [esi]; 현재 arr[i]
22         cmp eax, max_value
23         jle skip_update; max보다 작으면 건너뛰기
24
25         mov max_value, eax; max 갱신
26
27         skip_update :
28         add esi, 4; 다음 요소
29         dec ecx
30         jmp loop_start
31         end_loop :
32         jmp done
33         no_elements :
34         mov max_value, 0; 요소 없으면 0 반환
35         done :
36     }
37     return max_value;
38 }

```

출력

출력 보기 선택(S): 빌드

1>----- 빌드 시작: 프로젝트: Max_dll, 구성: Debug Win32 -----

1>maxdll.c

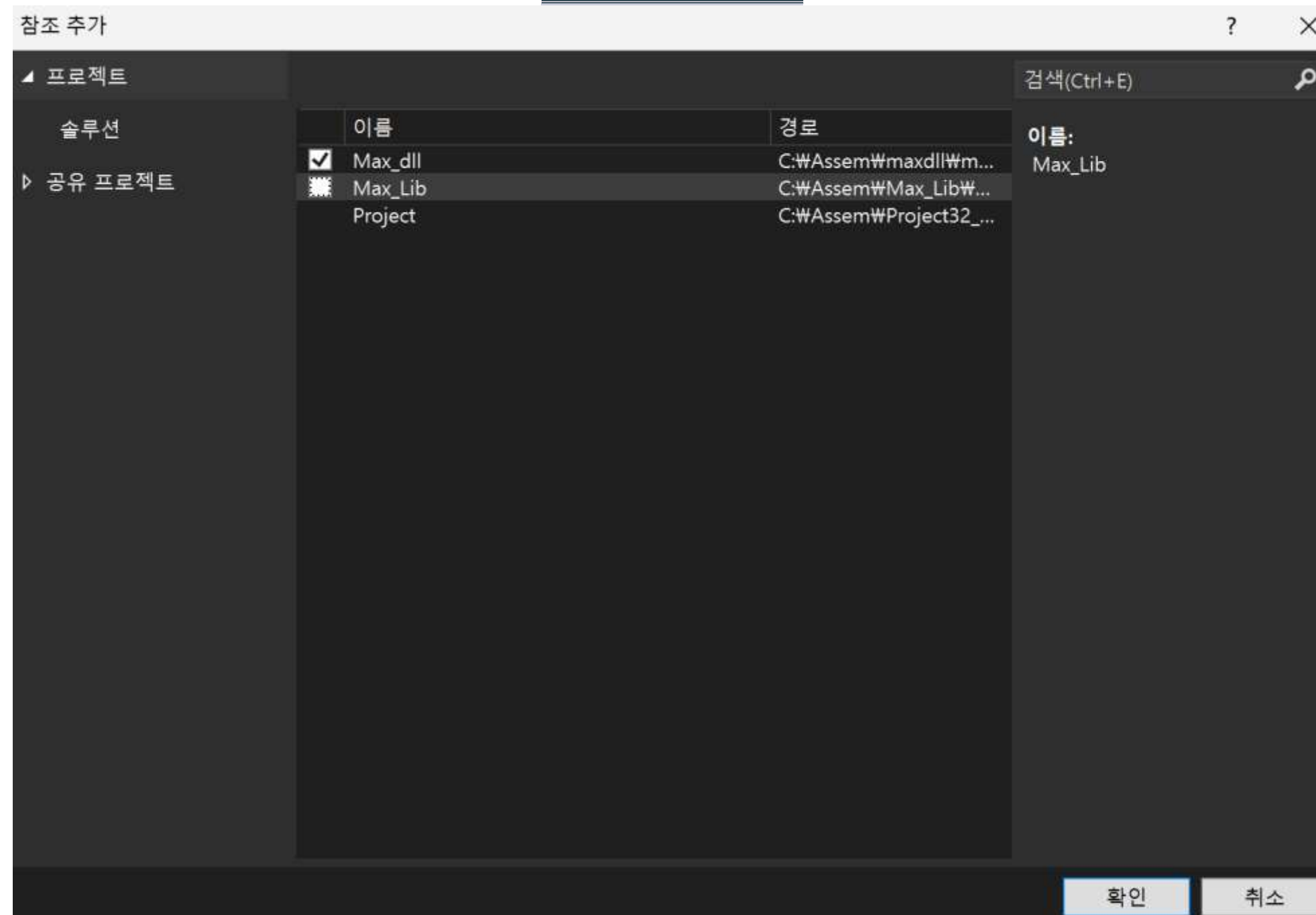
1> C:\Assem\Project32_VS2022\Debug\Max_dll.lib 라이브러리 및 C:\Assem\Project32_VS2022\Debug\Max_dll.exp 개체를 생성하고 있습니다.

1>maxdll.vcxproj -> C:\Assem\Project32_VS2022\Debug\Max_dll.dll

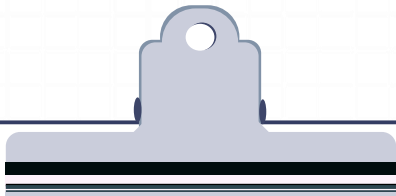
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛸 =====

===== 빌드(가) 오전 1:49에 완료되었으며, 01.353 초(가) 걸림 =====

기존의 함수(maxlib.c에 사용한 코드) 앞에 __declspec(dllexport)를 추가하여 Max_dll을 작성하고 빌드를 하여 dll파일을 생성해줍니다.



라이브러리 때와 마찬가지로 실행할 프로젝트인 array_max의 참조에 방금 생성한 Max_dll을 추가해줍니다.



```
maxdll.c  maxlib.c  main.c  X
array_max  (전역 범위)
1  #include <stdio.h>
2
3  __declspec(dllexport) int find_max(int arr[], int len);
4
5  int main()
6  {
7      int arr[5];
8
9      printf("정수 5개를 입력하세요:\n");
10     for (int i = 0; i < 5; i++)
11         scanf_s("%d", &arr[i]);
12
13     int max_val = find_max(arr, 5);
14
15     printf("가장 큰 값: %d\n", max_val);
16
17     return 0;
18
19
```

Microsoft Visual Studio 디버그 X + v - □ X

정수 5개를 입력하세요 :
3
6
9
729
20
가장 큰 값 : 729

C:\Assem\Project32_VS2022\Debug\array_max.exe(프로세스 44084)이(가) 0 코드(0x0)와 함께 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요...

그리고 실행할 파일인 main.c에 있는 함수 코드 앞에도 `__declspec(dllexport)`를 작성해줍니다. 정상적으로 작동하는 모습을 볼 수 있습니다.

점수

10/10

교수님께서 제시해주신
내용들을 모두 수행하여
10점으로 채점하였습니다.



참고자료

<https://chatgpt.com/>

<https://gemini.google.com/app?hl=ko>