C Programming

Practice 2

How to use printf

- Output format character

Format character	Output type
%d, %i	**decimal integer**
%x, %o	Hexadecimal integer, Octal integer
%f, %lf	**Float type real number/double type real number**
%c	**A character**
%s	**String**
%u	Decimal integer(positive only)
%%	% symbol

- Ex)
  - int a = 3;
  - printf (“%d”, a);

How to use scanf

- input format character

Format character	Input type
%d	**decimal integer**
%x, %o	Hexadecimal integer, Octal integer
%f, %lf	**Float type real number/double type real number**
%c	**A character**
%s	**String**
%u	Decimal integer(positive only)

- Ex)
  - int a;
  - scanf (“%d”, &a); <& : address operator>

While statement structure

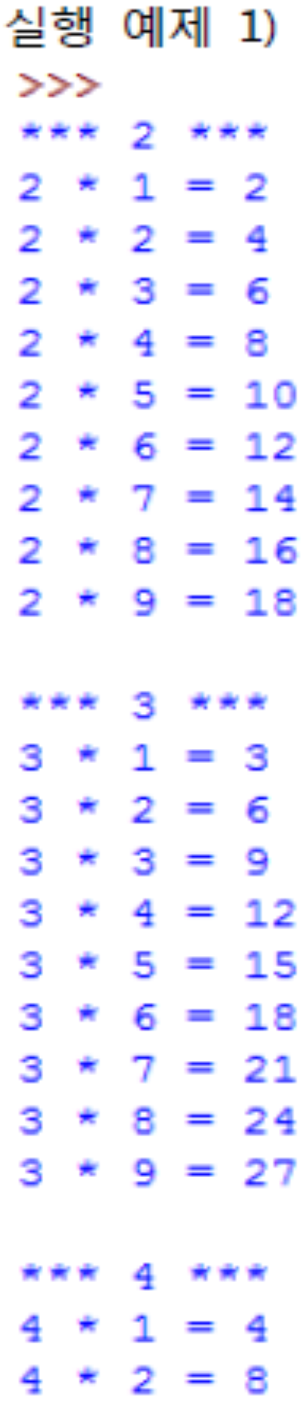
```
int i = 0; // variable for while statement.

while( i < 10 ) // i < 10 is a conditional sentence.
{
        printf("%d",i); // print the i value.
        i = i + 1; // change the i value.
}
```

Do-While statement structure

```
do{

    1. statements
    2. Increment or Decrement statement

} while ( condition statement  );
```

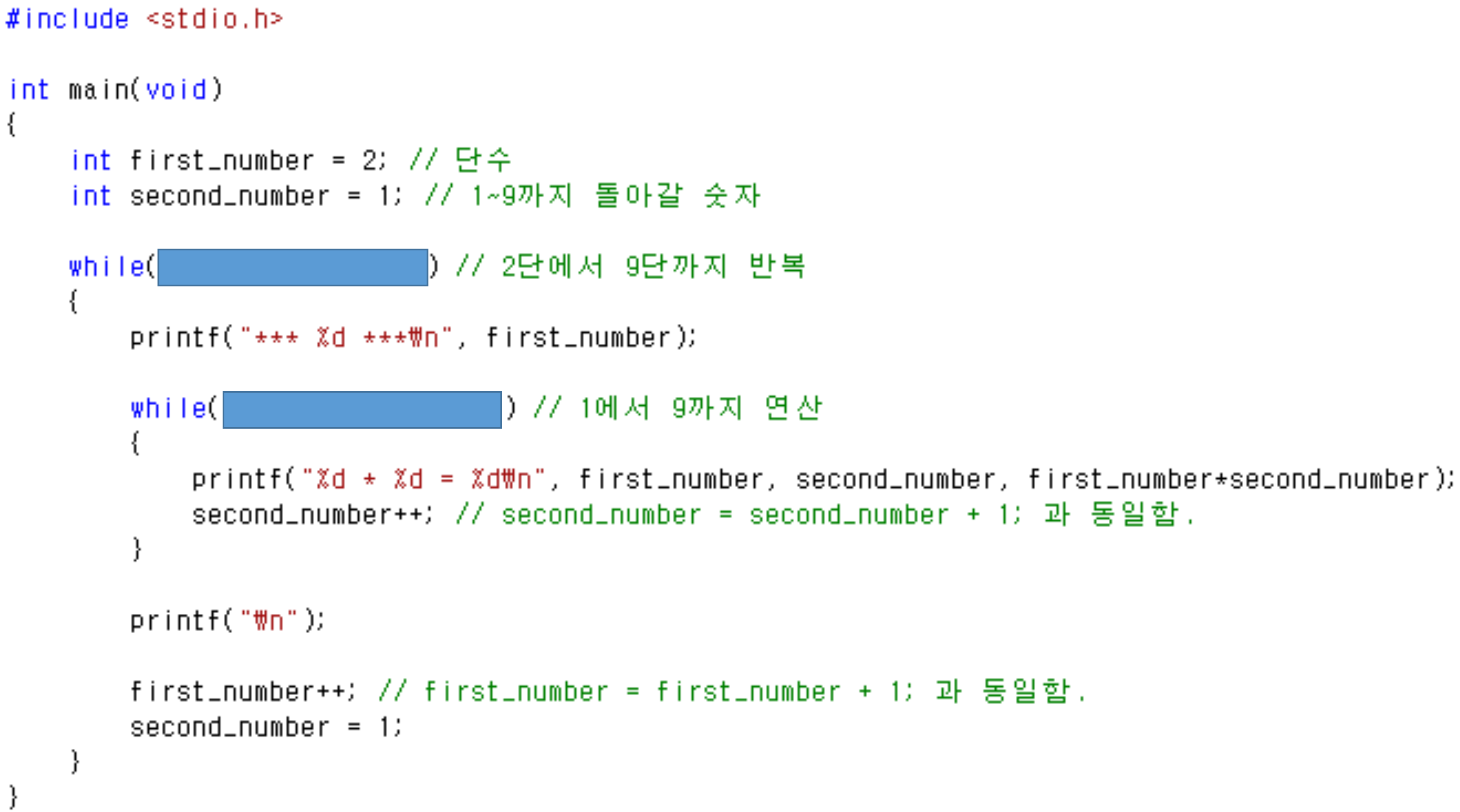
Homework 3 - Nested while

- Output multiplication tables from step 2 to step 9
- Use the nested while statement
- Don’t use printf function more than 4 times



실행 예제 1)

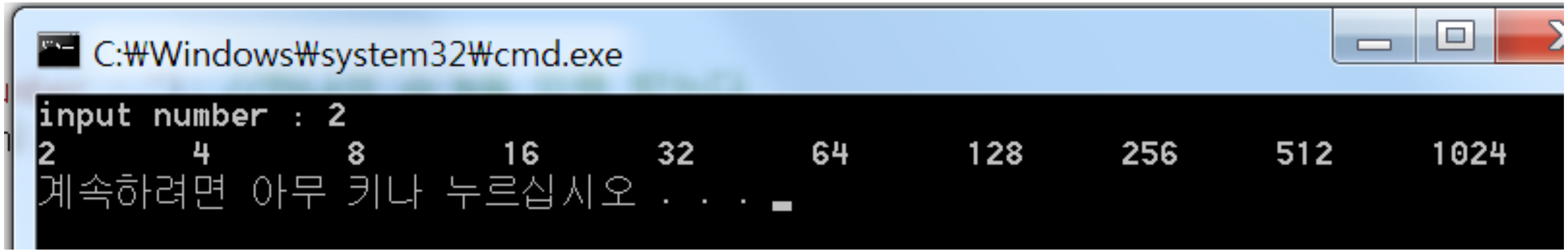
```
>>>
*** 2 ***
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18

*** 3 ***
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

*** 4 ***
4 * 1 = 4
4 * 2 = 8
```



```
#include <stdio.h>

int main(void)
{
    int first_number = 2; // 단수
    int second_number = 1; // 1~9까지 돌아갈 숫자

    while(                    ) // 2단에서 9단까지 반복
    {
        printf("*** %d ***₩n", first_number);

        while(                    ) // 1에서 9까지 연산
        {
            printf("%d * %d = %d₩n", first_number, second_number, first_number*second_number);
            second_number++; // second_number = second_number + 1; 과 동일함.
        }

        printf("₩n");

        first_number++; // first_number = first_number + 1; 과 동일함.
        second_number = 1;
    }
}
```

Homework 4 - computing powers of n

- Input a number N
- Output powers of N using the while statement
- 10 outputs
- Ex)



```
input number : 2
2       4       8       16      32      64      128     256     512     1024
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int exponent = 0, power_of_n = 1;
    int num = 0;

    printf("input number : "); //하나의 수 N을 입력 받는다.
    scanf("%d", &num);

    while (++exponent <= 10)
    {
        printf("%d₩t",                    );
    }
    printf("₩n");
    return 0;
}
```

Homework form

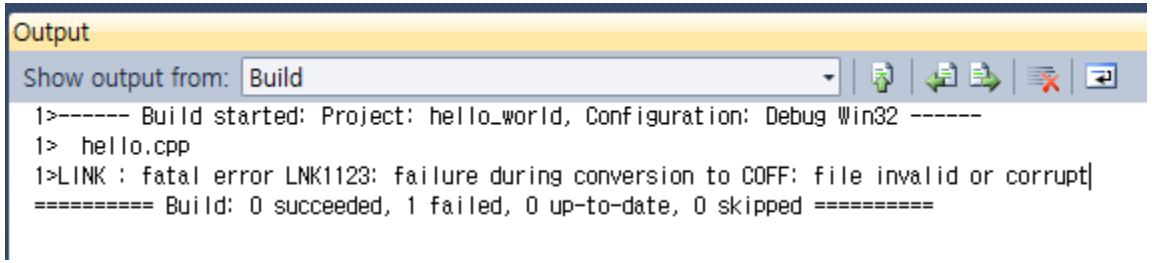
- Homework submission e-mail:

  hizorro99@naver.com

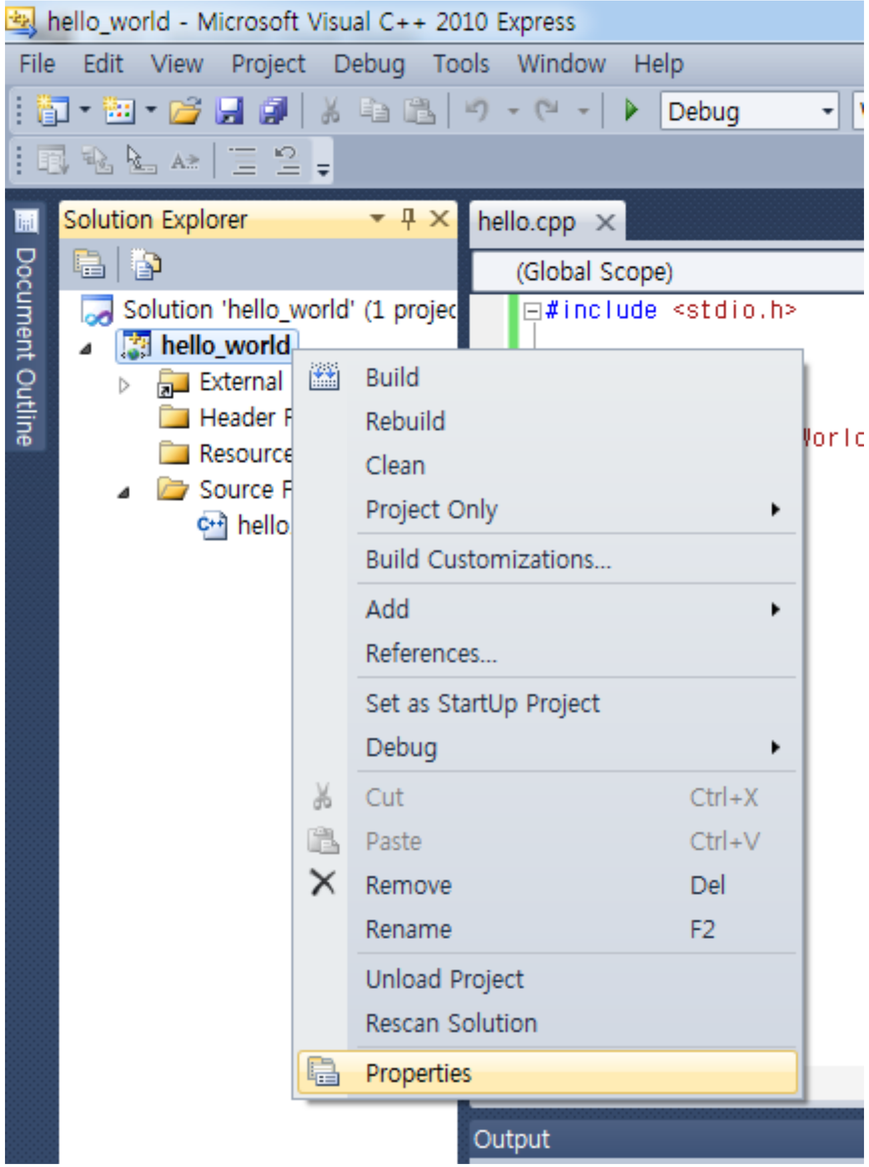
- E-mail title: day(Thursday or Friday)_name_#week
  - Ex) Friday_james_week3
  - Ex) 목요일반_장원철_3주차
- File title: student id_name_week#_1.c
  - Ex) 2014123456_james_week3_1.c (or .cpp)
  - Ex) 2014123456_james_week3_2.c (or .cpp)

예외

- 이러한 오류가 뜨는 경우

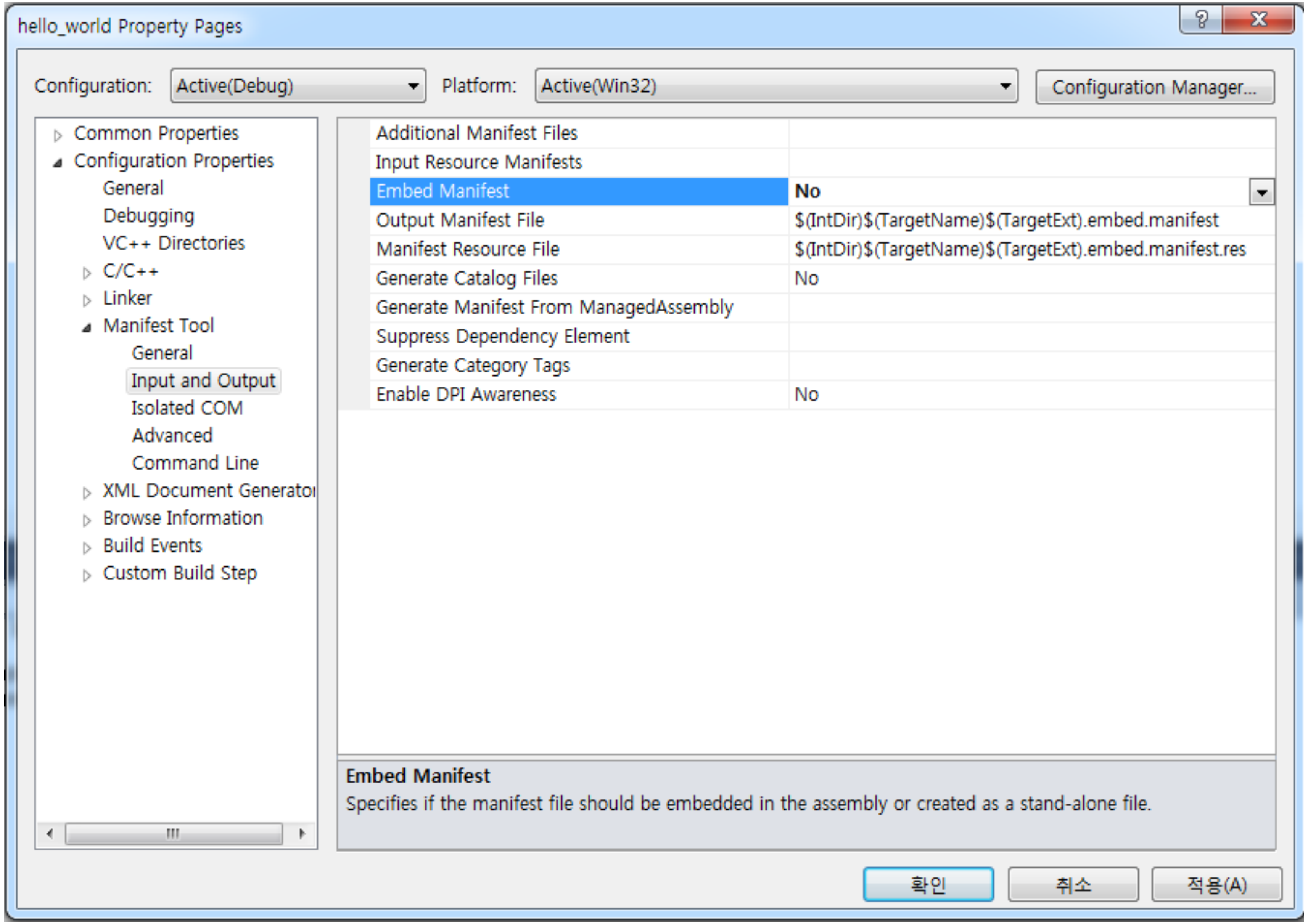


```
1>------ Build started: Project: hello_world, Configuration: Debug Win32 ------
1>  hello.cpp
1>LINK : fatal error LNK1123: failure during conversion to COFF: file invalid or corrupt
========== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped ==========
```



```
#include <stdio.h>
```

hello_world - Microsoft Visual C++ 2010 Express
File
Edit
View
Project
Debug
Tools
Window
Help
Debug
Document Outline
Solution Explorer
Solution 'hello_world' (1 projec
hello_world
External
Header F
Resource
Source F
hello
hello.cpp
(Global Scope)
Vorlc
Build
Rebuild
Clean
Project Only
Build Customizations...
Add
References...
Set as StartUp Project
Debug
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename F2
Unload Project
Rescan Solution
Properties
Output



Configuration: Active(Debug)

Platform: Active(Win32)

Additional Manifest Files	
Input Resource Manifests	
Embed Manifest	**No**
Output Manifest File	$(IntDir)$(TargetName)$(TargetExt).embed.manifest
Manifest Resource File	$(IntDir)$(TargetName)$(TargetExt).embed.manifest.res
Generate Catalog Files	No
Generate Manifest From ManagedAssembly	
Suppress Dependency Element	
Generate Category Tags	
Enable DPI Awareness	No

Embed Manifest

Specifies if the manifest file should be embedded in the assembly or created as a stand-alone file.