

*8주차 : Longest Common Subsequence*

# 알 고 리 즈

2015. 10. 22.

충남대학교 컴퓨터공학과 임베디드 시스템 연구실  
TA 권진세

# Overview

## ▶ 이번 주 실습 / 과제

- 1) Dynamic Programming을 이용한  
Longest Common Subsequence 구현

# Longest Common Subsequence

## ▶ Dynamic Programming

- Programming은 테이블을 뜻함
  - > Divide and Conquer : disjoint로 분할 후 해를 구하고 결합
  - > Dynamic Programming : 문제가 중복될 때 반복되는 일을  
테이블로 만들어 계산의 중복을 피함

# Longest Common Subsequence

## ▶ Longest Common Subsequence

– LCS란?

\* 최장 공통 부분 시퀀스로서 연속된 문자의 유사도를 비교하는 알고리즘

– 활용 : DNA base(기저) 분석

S1 = ACCG**GTCG**AG**TGCG**CGAAGCCGGCCGAA

S1 = **GTCG****TCGGAATGCCGTTGCTCTGTAAA**

S1 = GTCGTCGGAAGCCGGCCGAA

# Longest Common Subsequence

## ▶ Longest Common Subsequence

– LCS란?

\* 최장 공통 부분 시퀀스로서 연속된 문자의 유사도를 비교하는 알고리즘

– 활용 : DNA base(기저) 분석

S1 = ACCG**GTCG**AG**TGCG**CGAAGCCGGCCGAA

S2 = **GTCG****TCGGAATGCCGTTGCTCTGTAAA**

S3 = GTCGTCGGAAGCCGGCCGAA

# Longest Common Subsequence



## Dynamic programming

*Design technique, like divide-and-conquer.*

### **Example: Longest Common Subsequence (LCS)**

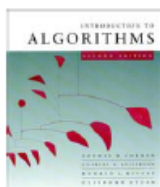
- Given two sequences  $x[1 \dots m]$  and  $y[1 \dots n]$ , find a longest subsequence common to them both.

“a” not “the”

$x$ : A B C B D A B

$y$ : B D C A B A

# Longest Common Subsequence



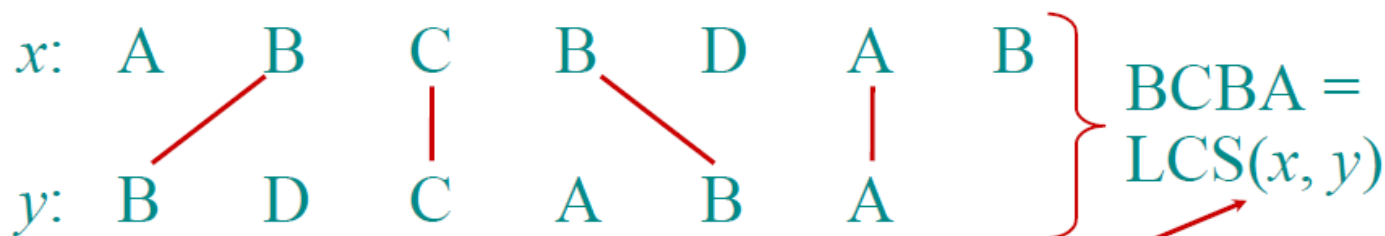
## Dynamic programming

*Design technique, like divide-and-conquer.*

### Example: *Longest Common Subsequence (LCS)*

- Given two sequences  $x[1 \dots m]$  and  $y[1 \dots n]$ , find a longest subsequence common to them both.

“a” not “the”



functional notation,  
but not a function

# Longest Common Subsequence

***Memoization:*** After computing a solution to a subproblem, store it in a table. Subsequent calls check the table to avoid redoing work.

$\text{LCS}(x, y, i, j)$

**if**  $c[i, j] = \text{NIL}$   Nothing

**then if**  $x[i] = y[j]$

**then**  $c[i, j] \leftarrow \text{LCS}(x, y, i-1, j-1) + 1$

**else**  $c[i, j] \leftarrow \max \{ \text{LCS}(x, y, i-1, j), \text{LCS}(x, y, i, j-1) \}$

*same  
as  
before*



# Longest Common Subsequence

		A	B	C	B	D	A	B
	0	0	0	0	0	0	0	0
B	0							
D	0							
C	0							
A	0							
B	0							
A	0							

# Longest Common Subsequence

	A	B	C	B	D	A	B
0	0	0	0	0	0	0	0
B	0						
D	0						
C	0						
A	0						
B	0						
A	0						

# Longest Common Subsequence

$A \neq B$

	A	B	C	B	D	A	B
0	0	0	0	0	0	0	0
B	0						
D	0						
C	0						
A	0						
B	0						
A	0						

# Longest Common Subsequence

		A	B	C	B	D	A	B
	0	0	0	0	0	0	0	0
B	0	$\leftarrow \text{Max}(0,0)$						
D	0							
C	0							
A	0							
B	0							
A	0							

# Longest Common Subsequence

B=B

	A	B	C	B	D	A	B
B	0	0	0	0	0	0	0
D	0						
C	0						
A	0						
B	0						
A	0						

# Longest Common Subsequence

B=B    A    **B**    C    B    D    A    B

	0	0	0	0	0	0	0
<b>B</b>	0	0	← 0 + 1				
D	0						
C	0						
A	0						
B	0						
A	0						

# Longest Common Subsequence

		A	B	C	B	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1					
D	0							
C	0							
A	0							
B	0							
A	0							

# Longest Common Subsequence

		A	B	C	B	D	A	B
		0	0	0	0	0	0	0
B		0	0	1	1	1	1	1
D		0	0	1	1	1	2	2
C		0	0	1	2	2	2	2
A		0	1	1	2	2	3	3
B		0	1	2	2	3	3	4
A		0	1	2	2	3	4	4



# Longest Common Subsequence

	A	B	C	B	D	A	B
B	0	0	0	0	0	0	0
D	0	0	1	1	1	2	2
C	0	0	1	2	2	2	2
A	0	1	1	2	2	3	3
B	0	1	2	2	3	3	4
A	0	1	2	3	3	4	4

# Longest Common Subsequence

## LCS-LENGTH (X,Y)

$m = X.length$

$n = Y.length$

$c[0...m, 0...n]$  새로운 테이블을 만든다.

for  $i = 0$  to  $m$

$c[i, 0] = 0$

for  $j = 0$  to  $n$

$c[0, j] = 0$

for  $i = 1$  to  $m$

    for  $j = 1$  to  $n$

        if  $x_i == y_j$

$c[i, j] = c[i - 1, j - 1] + 1$

        elseif  $c[i - 1, j] \geq c[i, j - 1]$

$c[i, j] = c[i - 1, j]$

        else

$c[i, j] = c[i, j - 1]$

return  $c$

# Longest Common Subsequence

**PRINT-LCS** (  $c$ ,  $X$ ,  $i$ ,  $j$  )

if  $i == 0$  or  $j == 0$

return  $c$

if  $c[i,j]-1 == c[i-1,j-1]$  &&  $c[i,j]-1 == c[i,j-1]$  &&  $c[i,j]-1 == c[i-1,j]$

**PRINT-LCS** (  $c$ ,  $X$ ,  $i$ ,  $j - 1$  )

$x_i$  를 출력한다.

elseif  $c[i,j] == c[i,j-1]$

**PRINT-LCS** (  $c$ ,  $X$ ,  $i$ ,  $j - 1$  )

else

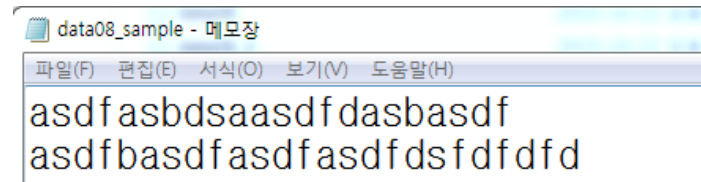
**PRINT-LCS** (  $c$ ,  $X$ ,  $i$ ,  $j - 1$  )

# Practice / Homework

## 1. Longest Common Subsequence 구현

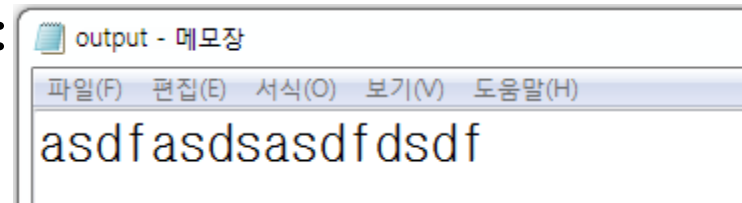
과제 예시)

Input Data :



```
data08_sample - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
asdfasbdsaasdfsasbasdf
asdfbasdfasdfsasdfsdfdfd
```

Output Data :



```
output - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
asdfsasdfsdf
```

# Practice / Homework

※ 그 외 실습 과제 수행 중 유의 사항

- 포함내용 : 코드만 제출

※ 이번주 보고서 제출을 없습니다.

※ 201500000\_08.c 파일만 전송

(파일이 여러 개인 경우는 압축해서 보내세요)

- 제출이름 :

메일 : [알고리즘00반]\_201500000\_홍길동\_8주차

파일 : 201500000\_05.c

- 제출기한 : 2015-10-29 18:00까지

- 메일주소 : [kwonse@cnu.ac.kr](mailto:kwonse@cnu.ac.kr)

# APPENDIX 1. File I/O

## 1. 파일 입출력 방법

`FILE* fp;           //fp : input file pointer`

`FILE* fop;          //fop : input file pointer`

`//파일 이름은 "00_201500000_insertion.txt"`

`//입출력 파일은 *.c 소스파일과 같은 폴더에 있어야 한다.`

`fp = fopen(FILENAME,"rt");           //입력 파일 열기`

`fop = fopen(FILENAME2,"wt");         //출력 파일 열기`

`if (fp == NULL) {`

`printf("**** Input File open error ****\n");`

`exit(1);`

`} //파일 없을 경우 예외처리로 프로그램 종료`

# APPENDIX 1. File I/O (계속)

```
while(!feof(fp)){  
} //파일의 끝날때 까지 반복
```

```
fscanf(fp,"%d, ", &변수); // ex) 123, 456,  
// int 값만 추출함 ', '는 제외됨
```

```
fprintf(fop, "%d", 출력할 값); // 파일 출력 시 사용
```

```
fclose(fp);
```

# APPENDIX 2. 배열 넘기기

1. Main 함수의 배열을 주소로 넘겨서 다룰 때 !! 이중포인터 사용  
- 장점 : 메모리 절약, 리턴 불필요.

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

```
void user_malloc(int** num);
```

```
void main(void){
    int *ptr;
    user_malloc(&ptr);    //포인터 변수 ptr의 주소를 인자로 보냄.
    printf("%d\n", *ptr); //출력 값은 10 이다.
    return 0;
}
```

```
void user_malloc(int** num){
    *num = (int*)malloc(sizeof(int));
    (*num)[0] = 10;
}
```



# APPENDIX 3. 동적 할당 메모리 크기

## Q & A.

### 포인터로 받은 배열의 크기를 구하는 방법? (있다)

- Malloc 함수의 선언을 보면 `void* malloc(size_t size)`
- `Size_t`는 많은경우 `unsigned long int`로 되어있으므로
- 메모리를 할당 할 때 이 크기만큼 더 할당해서 할당 영역의 처음 부분에 길이의 값을 저장해 두고 있음.

- `*(ptr - sizeof(size_t))`
- 다음과 같은 함수로 만들어 사용 가능

```
int sizeof_ar(int* S){  
    int size;  
    size = *(S - sizeof(int));  
    return size;  
}
```

# APPENDIX 4. 중간 값 찾기

3개의 원소중에 중간 값을 찾는 방법

```
int iPivot;  
int ptrCenter = (ptrLeft + ptrRight) / 2;  
if(!(ptrLeft < ptrCenter ^ ptrCenter < ptrRight))  
    iPivot = ptrCenter;  
else if(!(ptrCenter < ptrLeft ^ ptrLeft < ptrRight))  
    iPivot = ptrLeft;  
else  
    iPivot = ptrRight;
```

# APPENDIX 5. quick sort lib func

## Quick sort library function

**#include <stdlib.h>**

```
int compareX(const void* a, const void* b)
{
    d2_arr *p1 = (d2_arr *)a, *p2 = (d2_arr *)b;
    return (p1->x - p2->x);
}
int compareY(const void* a, const void* b)
{
    d2_arr *p1 = (d2_arr *)a, *p2 = (d2_arr *)b;
    return (p1->y - p2->y);
}
```

**qsort(arr, arr size, element size, **compare\_위에참조** )**

# APPENDIX 6. 각 자료형의 최대크기

## Variant limits 헤더

**#include <limits.h>**

**=> 정수형 변수의 최대값을 전처리 매크로로 저장한 헤더**

**#include <float.h>**

**=> ex) double 사이즈의 최대 크기를 알고 싶을 때**

**=> printf("%lf", DBL\_MAX);**

# APPENDIX 7. 2차원 배열 동적 할당

```
int input, i;
```

```
int **array = (int**)malloc(sizeof(int *)*input);
```

```
for(i=0; i<input; i++)
```

```
    array[i] = (int *)malloc(sizeof(int)*input);
```

```
//생성된 array[input][input] 을 사용
```

```
for(i=0; i<input; i++)
```

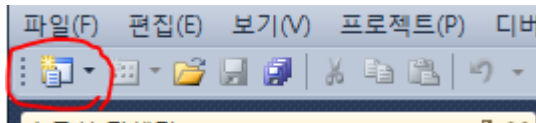
```
    free(array[i]);
```

```
free(array);
```

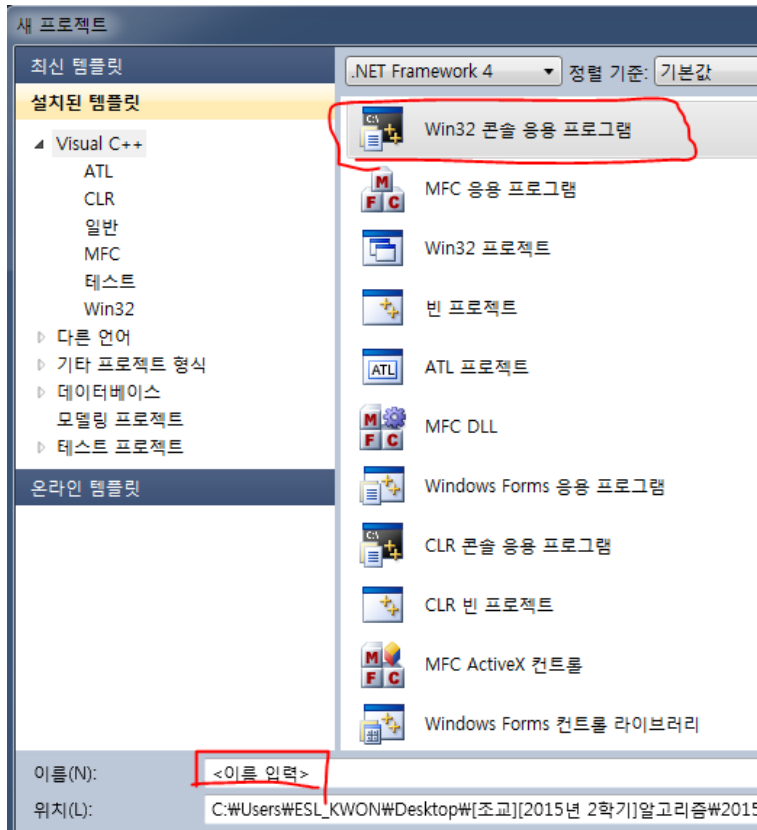
# Trouble Shooting

## Visual Studio 2010 프로젝트 생성

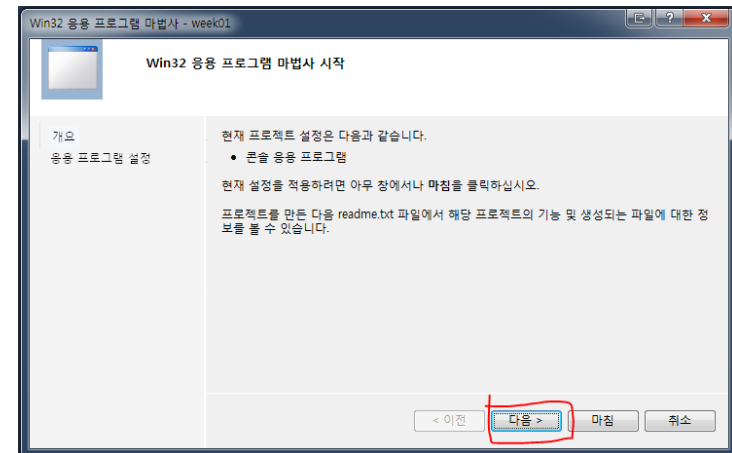
1.



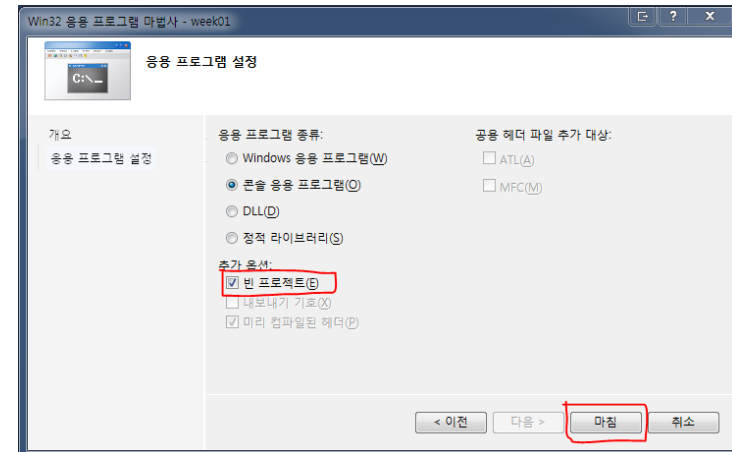
2.



3.

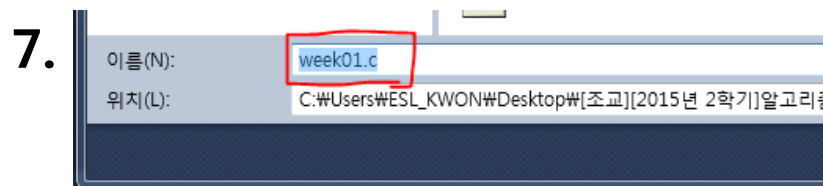
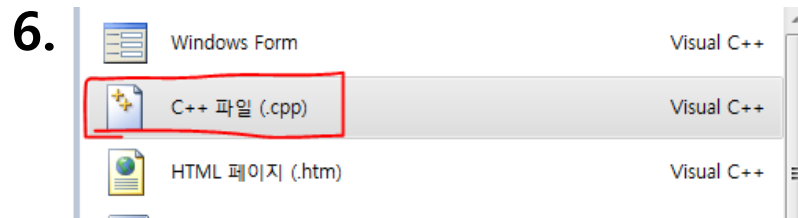
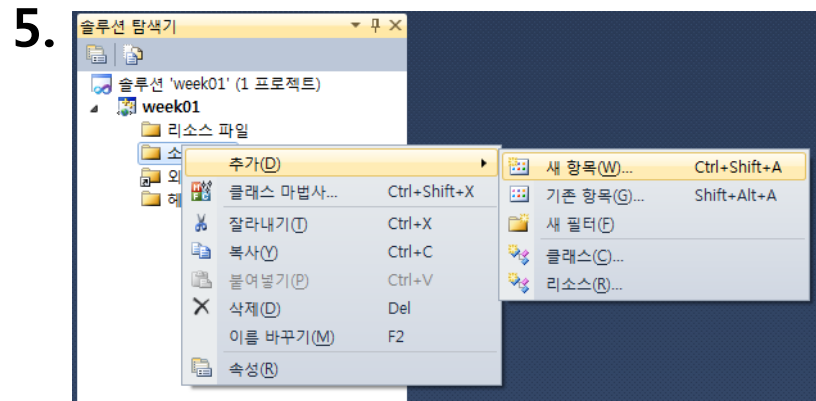


4.



# Trouble Shooting

## Visual Studio 2010 프로젝트 생성



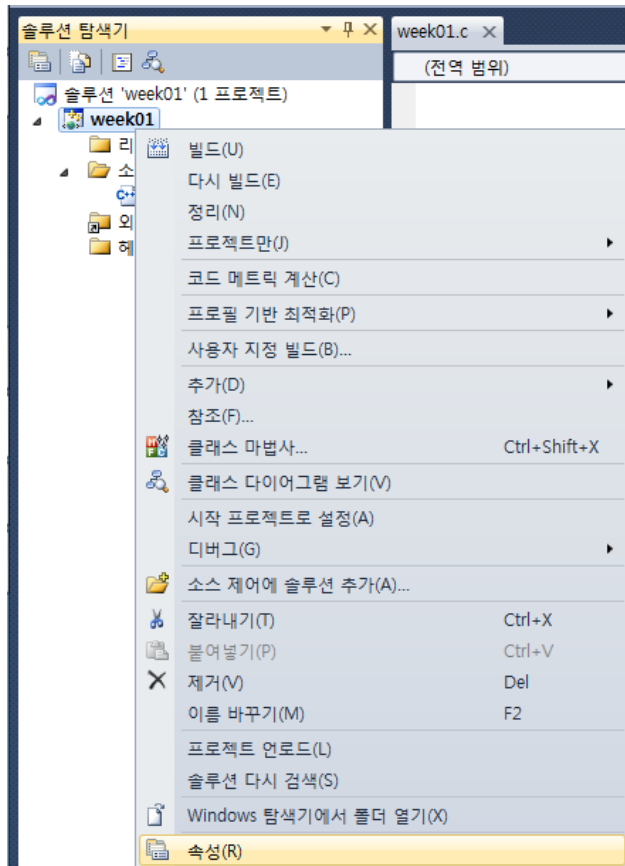
반드시 .c 로 이름 변경!!

# Trouble Shooting

## Visual Studio 2010 메니페스트 오류 해결

1>LINK : fatal error LNK1123: COFF로 변환하는 동안 오류가 발생했습니다. 파일이 잘못되었거나 손상되었습니다.  
1>  
1>빌드하지 못했습니다.

1.



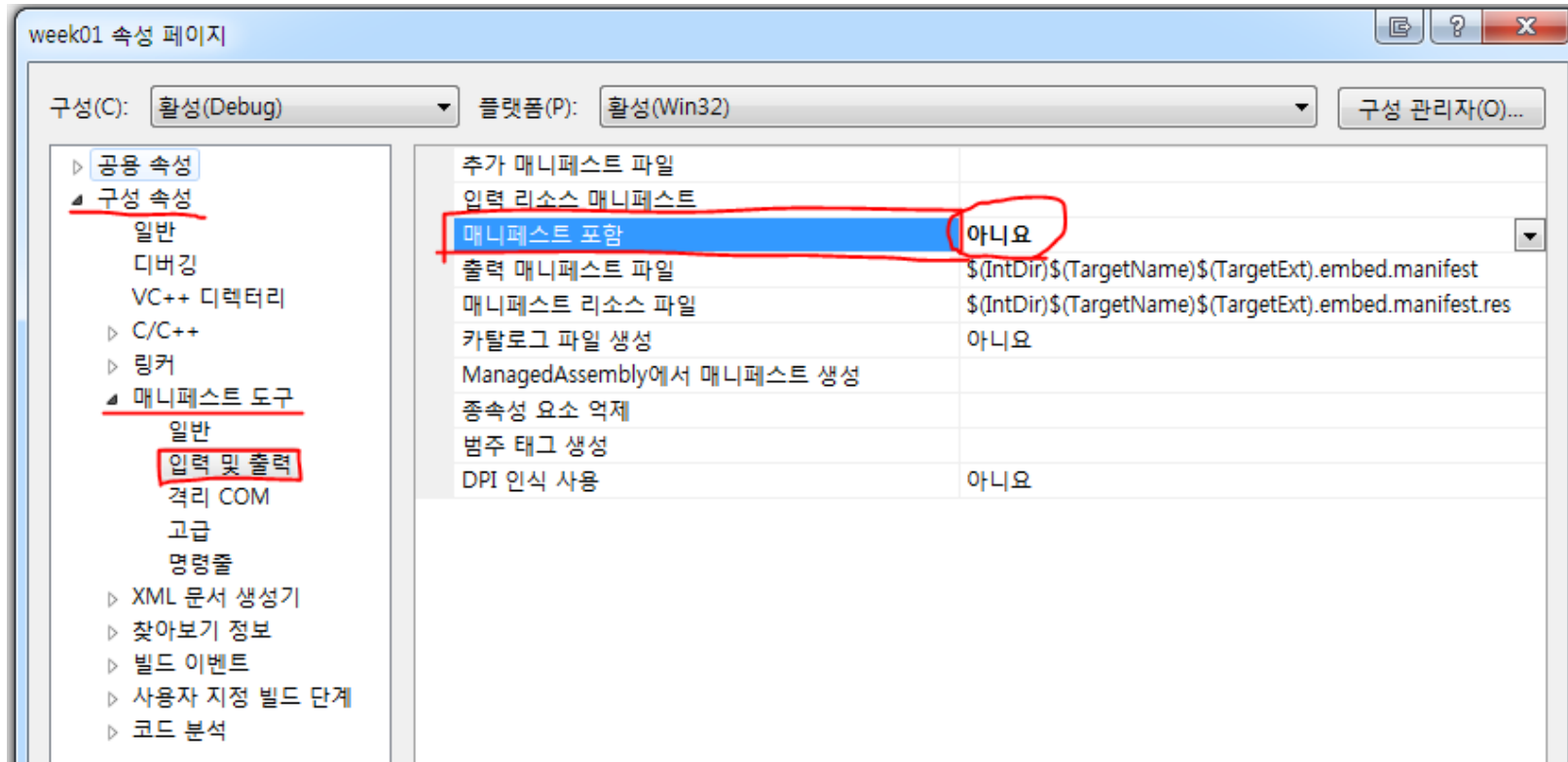
< 프로젝트 속성열기



# Trouble Shooting

## Visual Studio 2010 매니페스트 오류 해결

2.



구성 속성 -> 매니페스트 도구 -> 입력 및 출력 ->  
매니페스트포함 : "아니요 "

# Trouble Shooting

**Visual Studio 2010 매니페스트 오류 해결**

**3. 매니페스트 문제 영구적 해결 방법**

**Visual Studio Service Pack 1 다운로드.**

**( >600MB 오래 걸림... )**

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=23691>