

2008 Spring

# Computer Engineering Programming 1

## Lesson 1

- 1
- 2

Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

1

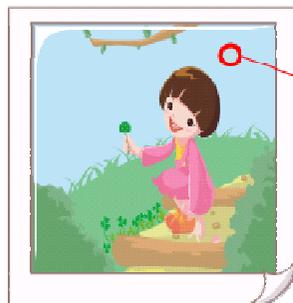
?

Q) (computer) 가?

A) (compute)

Q) . ? 가

A) .



각 화소의 밝기와 색상을 숫자로 표현한다

디지털 이미지의 경우



각 파형의 높이를 숫자로 표현한다.

디지털 음악의 경우

Q)

가?

A)

, , ,

Q)

가?

A)



오디오 테이프는 아날로그



mp3는 디지털

Q) 가 ?

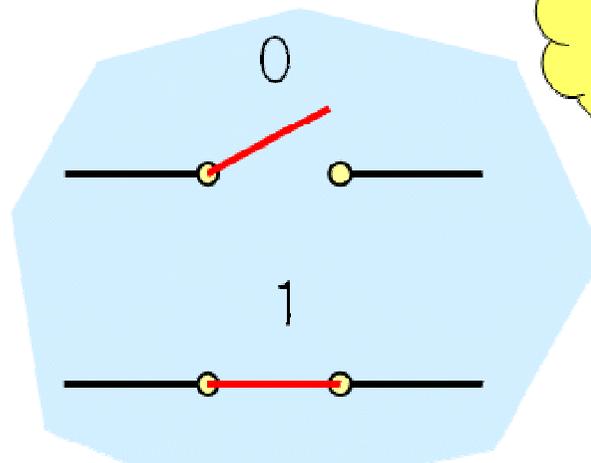
A)

Q)

A)

가 가

가



0은 열린 스위치로, 1은 닫힌 스위치로 표현할 수 있습니다.



Q)

가?

A)

0 1

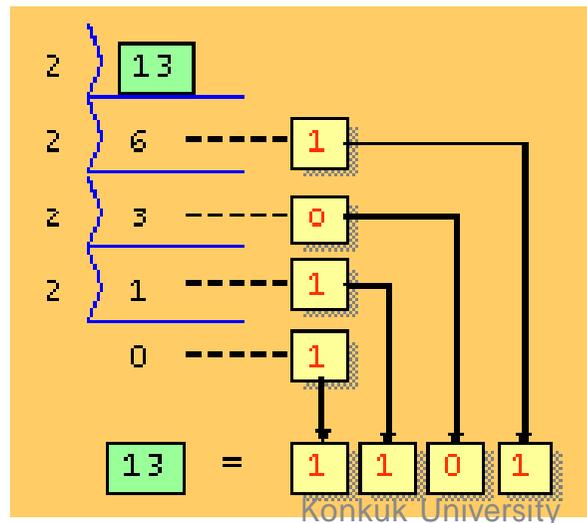
Q)

?

A)

0

2



Q) (bit) 가?

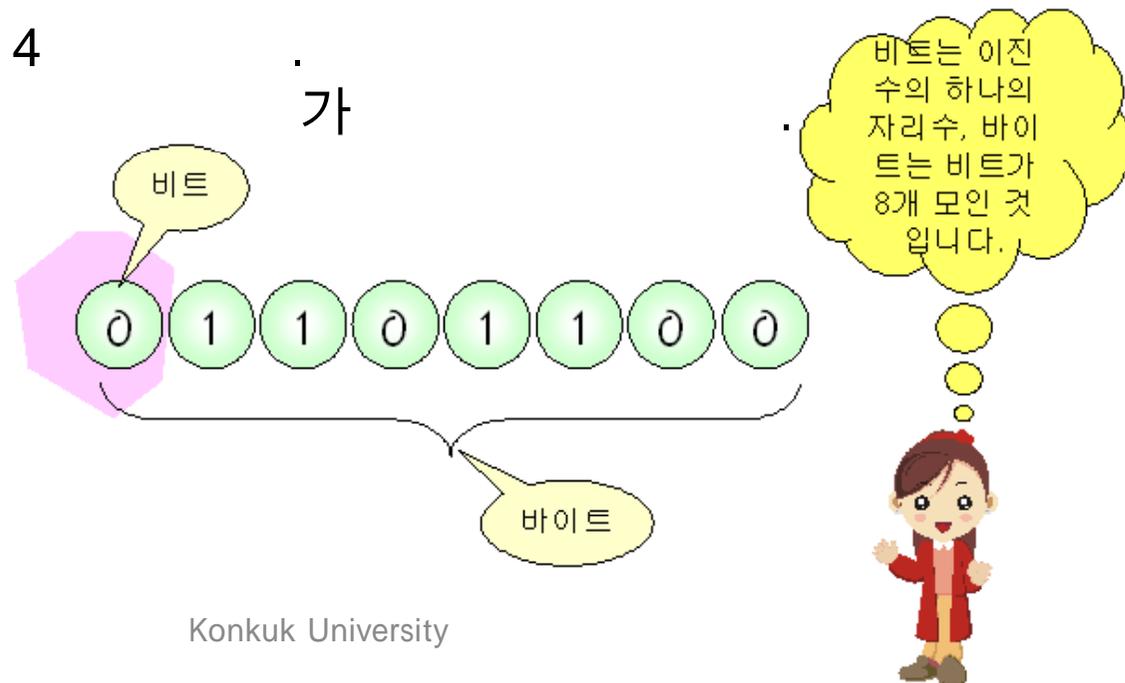
A) (binary digit)

Q) (byte) 가?

A) 8

Q) (word) 가?

A) 가 4



Q)

가?

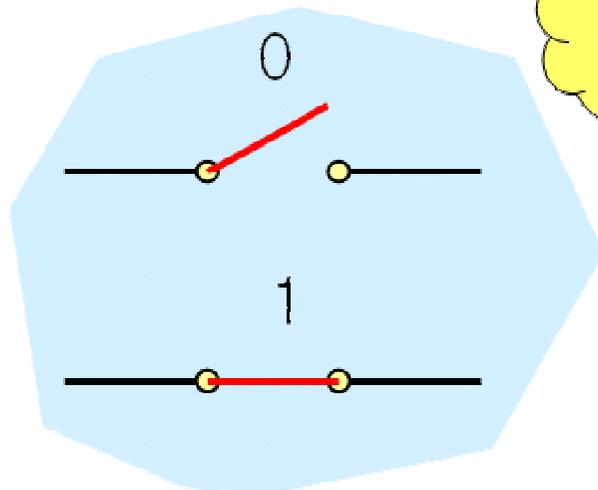
A)

0

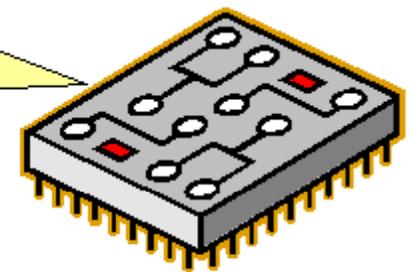
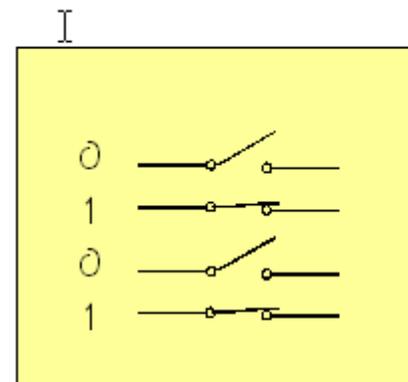
,

1

.



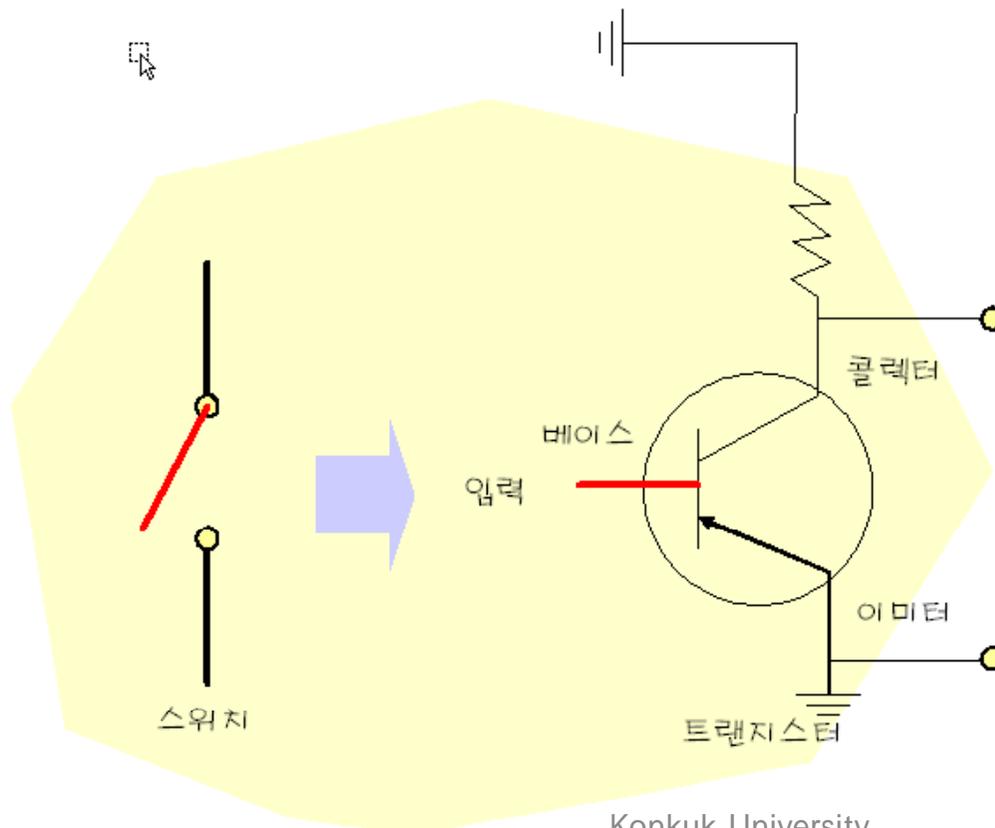
0은 열린 스위치로, 1은 닫힌 스위치로 표현할 수 있습니다.



Q)

가?

A)



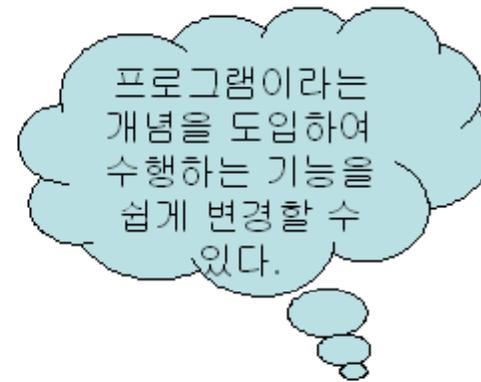
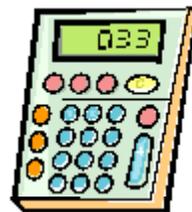
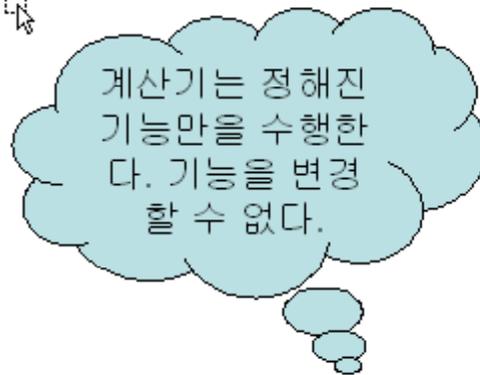
전압이 베이스에 걸릴 때만 전류를 흐르게 할 수 있습니다.



Q)

가?

A)



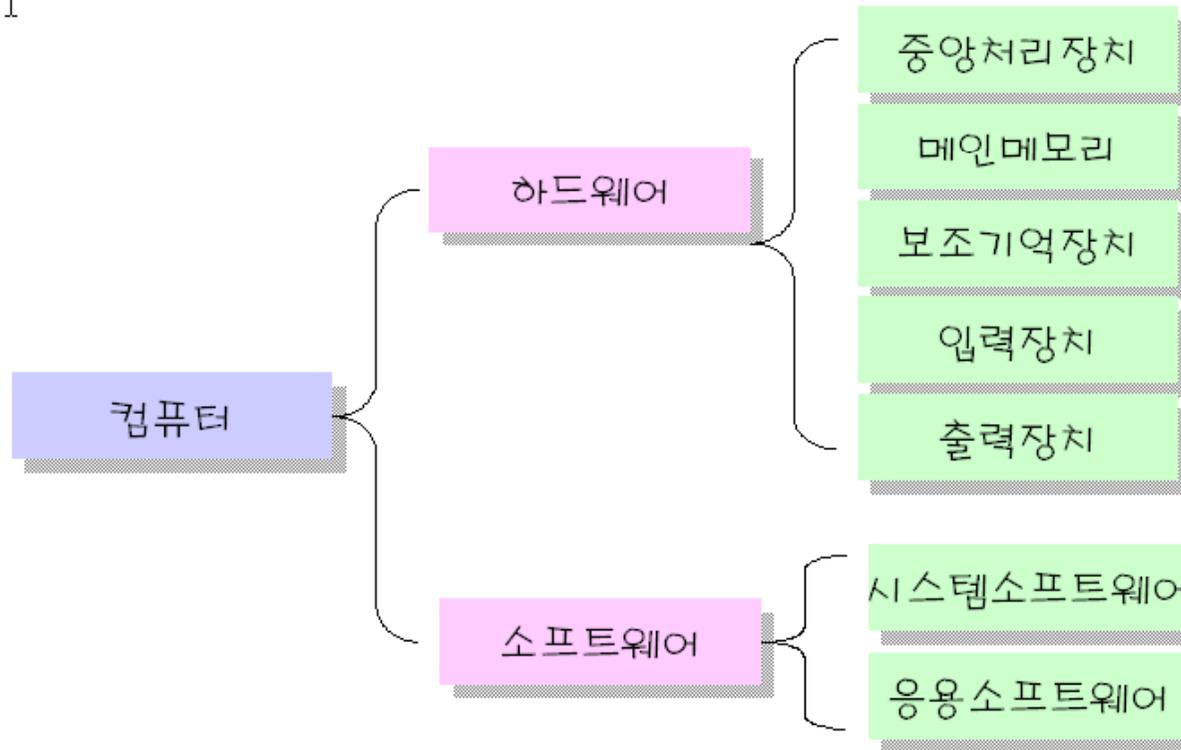
Q)

2가

?

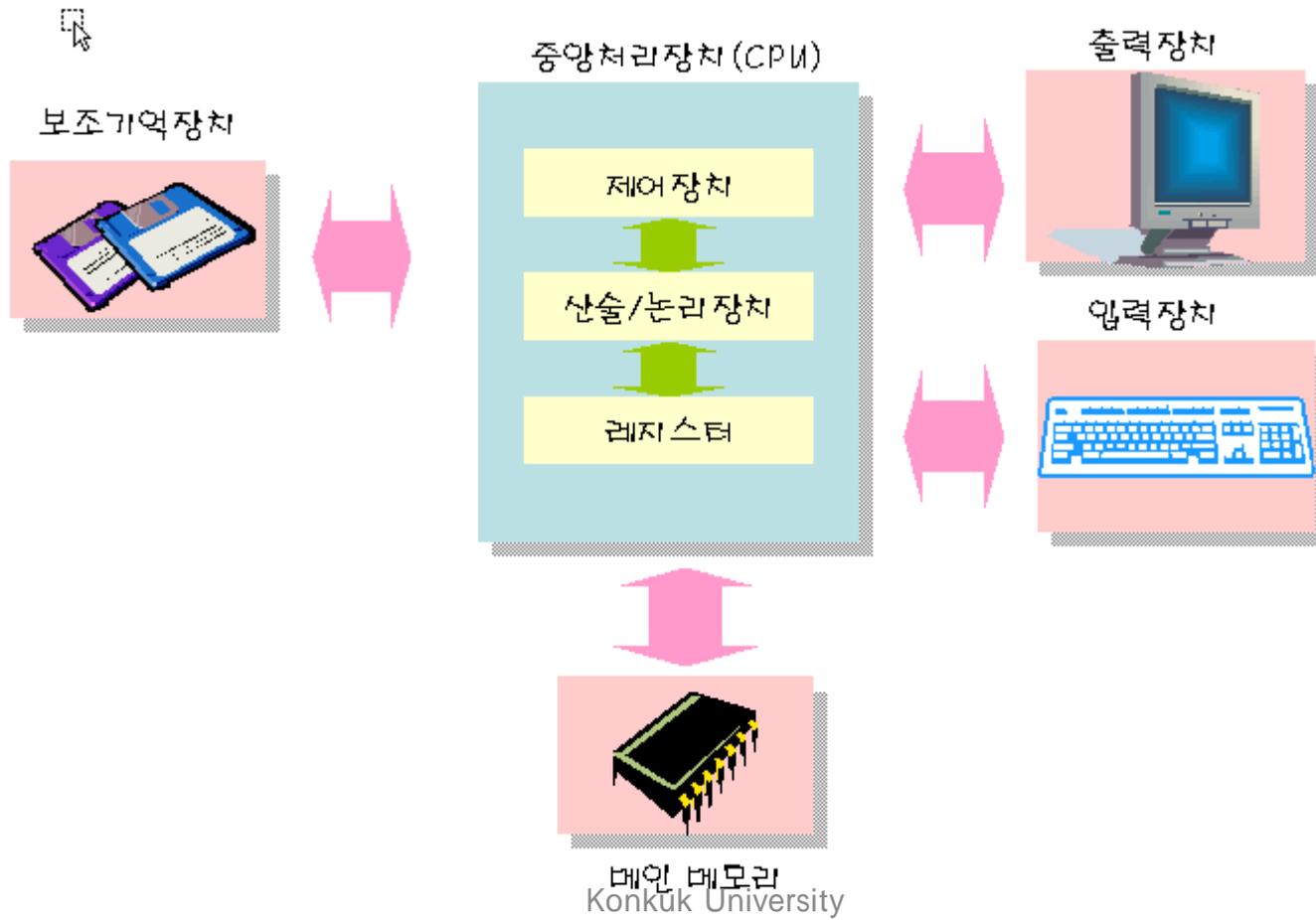
A)

I



Q)

가?



Q)

가?

소프트웨어

시스템 소프트웨어

: 운영체제, 컴파일러, 어셈블러, 링커, 로더, 프로그래밍 도구 등

응용 소프트웨어

: 워드 프로세서, 스프레드 시트, 그래픽 프로그램, 미디어 재생기 등

Q)

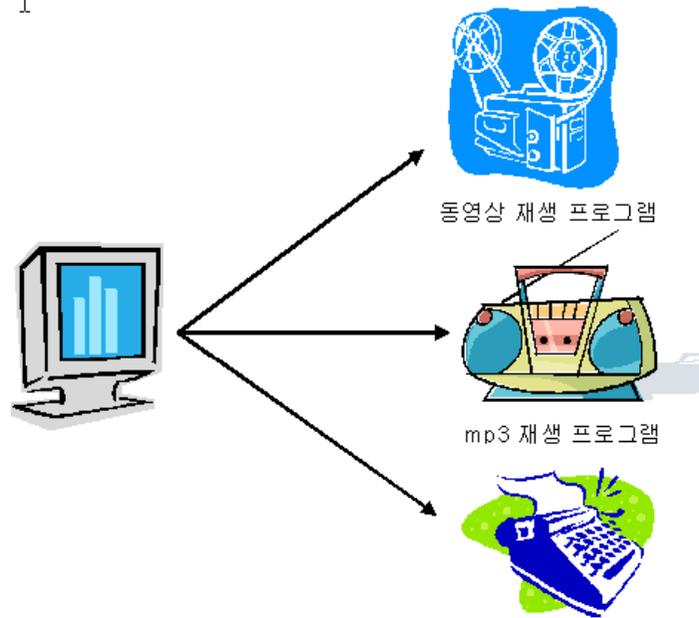
가

가

?

A)

I



Q)

가

가?

A)

가

가

0

1

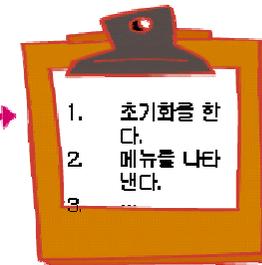
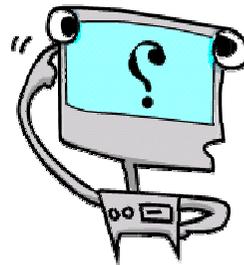
“001101110001010...”

A)

1

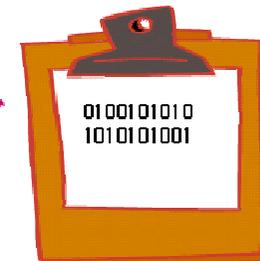
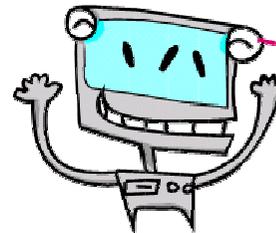
0

0  
1



한글로 된 작업지시서

ON/OFF



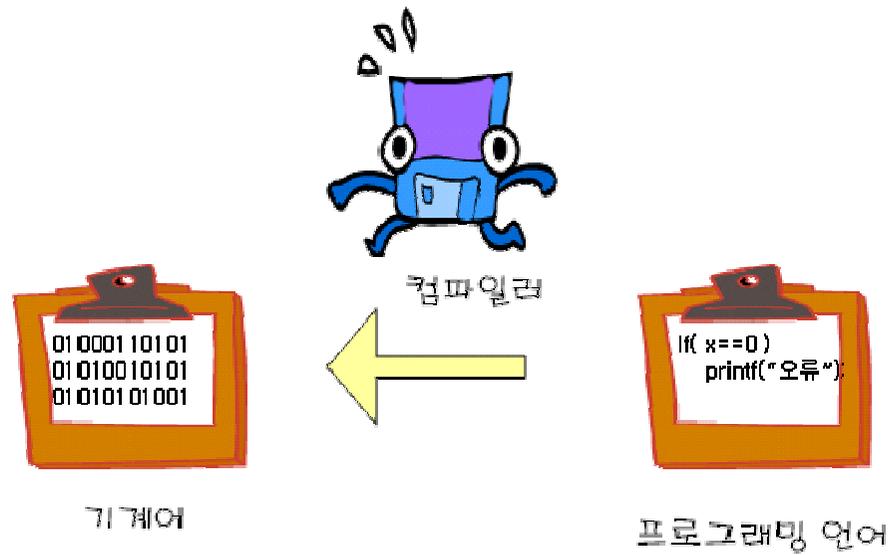
기계어로 된 작업지시서

Q)

- 
- 
- 

가

가?



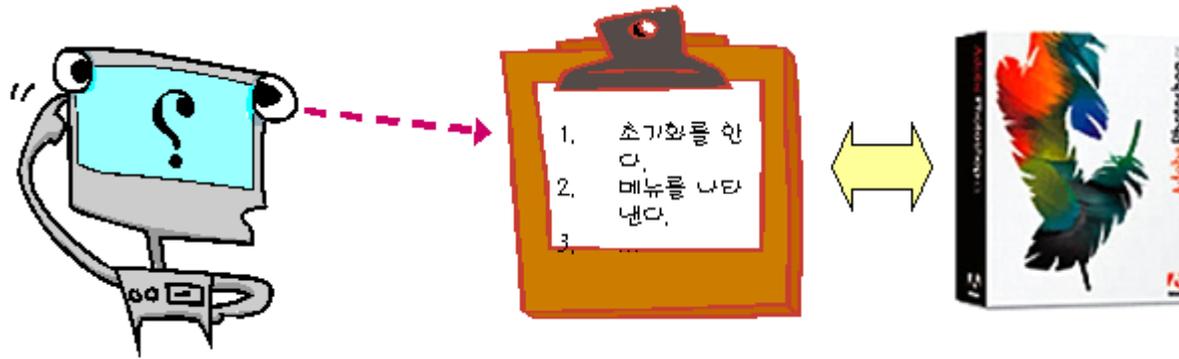
Q)

가?

A)

가

I



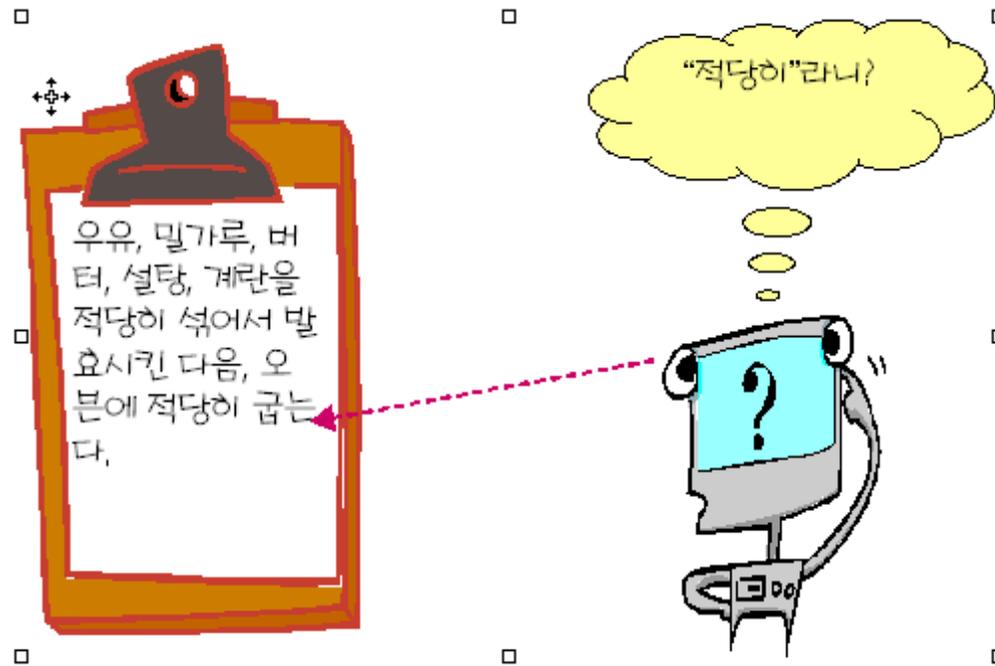
작업지시서

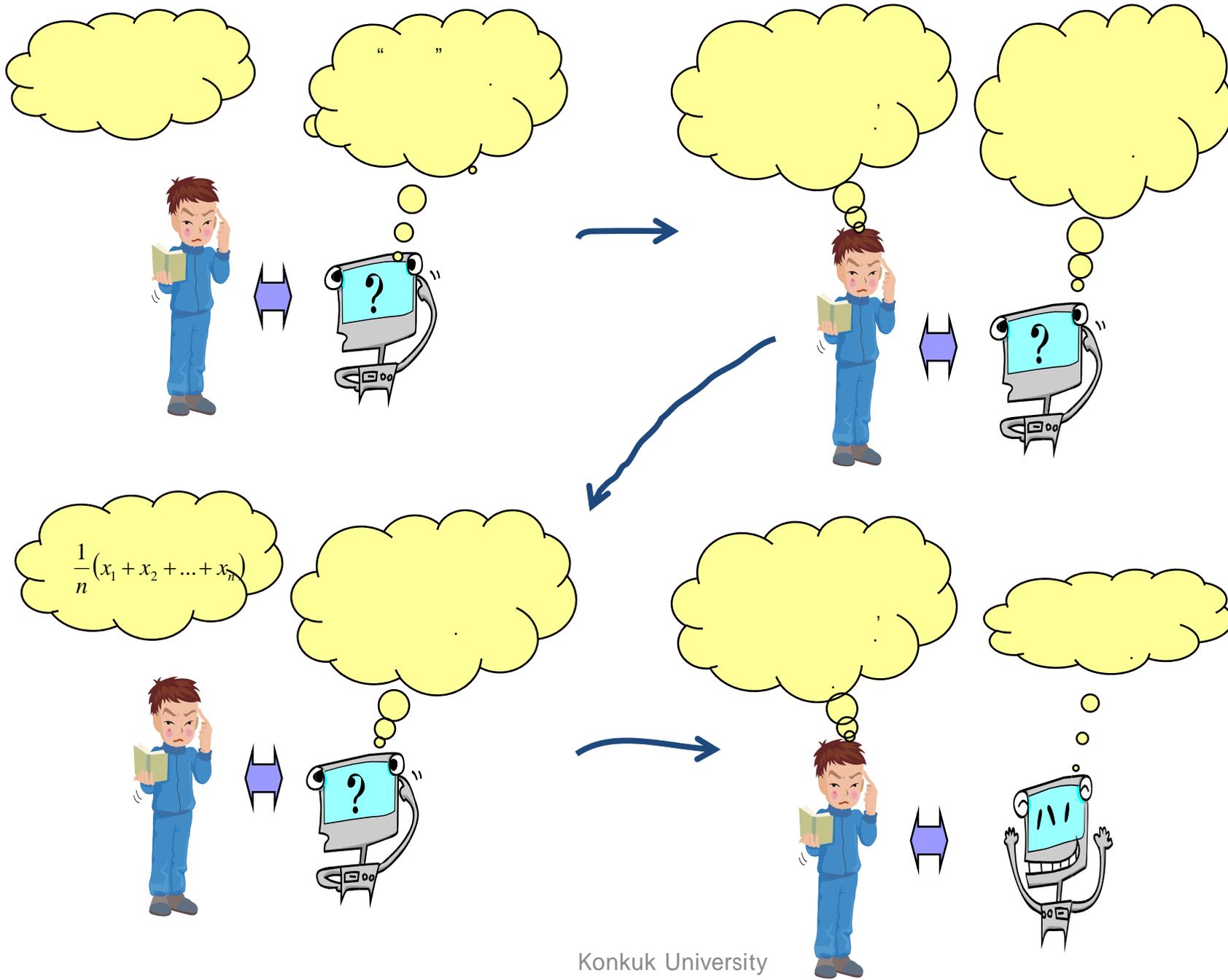
프로그램

Q)

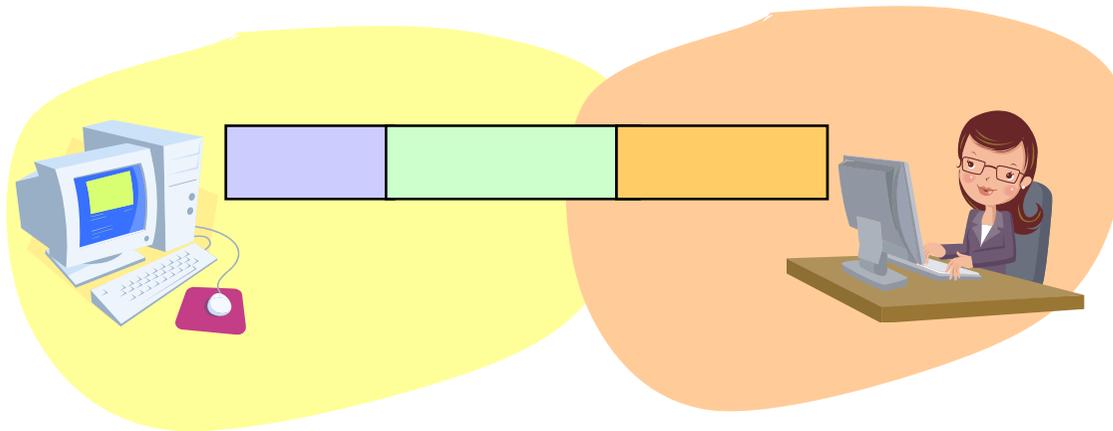
?

A)





- (machine language)
- (assembly language)
- (high-level language)



- 
- 0 1
- 

(instruction)

```
00001111 10111111 01000101 11111000
00001111 10111111 01001101 11111000
00000011 10100001
01100110 10001001 01000101 11111010
```

- CPU 가
- 가
- CPU 가
- (assembler):

```
MOV AX, MIDSCORE  
MOV CX, FINALSORE  
ADD AX CX  
MOV TOTALSCORE, AX
```

- ,
- C, C++, JAVA, FORTRAN, PASCAL
- :

```
TotalScore = MidScore + FinalScore;
```

# C

- 1970 AT&T Dennis Ritchie
- B ->C
- UNIX
- 가



*Ken Thomson Dennis  
Ritchie가  
National Medal of  
Technology*

# C

- K & R C
  - 1978 “C Programming Language”
  -
- ANSI C
  - 1983 ANSI(American National Standards Institute) X3J11
- C99
  - 1999 ISO
  - C++ 가 ( : C++ )
  -

C

- .
- .
- C  
가 .
- C  
가 .
- 가 가 .

# C

C 언어에는 꼭 필요한 기능만이 들어 있고 모든 표기법도 아주 간결하게 되어 있다.



간결하다.

C로 작성된 프로그램이 크기가 작으며 실행 속도가 빠르고 메모리를 효과적으로 사용한다. C언어는 거의 어셈블리 언어 수준의 효율성을 가진다.



효율적이다.

항상 모든 자유에는 책임이 따르듯이 하드웨어를 제어하기 위하여 꼭 필요한 요소인 포인터 등을 잘못 사용하는 경우가 많다.



배우기는 어렵다.

## C language



이식성이 뛰어나다.

한번 작성된 프로그램을 다른 CPU를 가지는 하드웨어로 쉽게 이식할 수 있다. PC에서 개발된 프로그램도 컴파일만 다시 하면 슈퍼 컴퓨터에서 수행시킬 수 있다.



저수준과 고수준이 모두 가능하다.

C 언어는 운영체제를 만들었던 언어이니 만큼, 어셈블리 언어 만큼의 구체적인 하드웨어 제어가 가능하다. 반면에 하향식 (top-down) 설계, 구조화 프로그래밍, 모듈화 설계 등의 소프트웨어 공학의 다양한 기법들을 적용할 수 있다.

# C

Q) C 가?

- C C++ JAVA
- C 가

*임베디드 시스템: 임베디드 시스템이란 특수 목적의 시스템으로 컴퓨터가 장치 안에 MP3 플레이어, 핸드폰등이 여기에 속한다.*



mp3 플레이어도 CPU와 플래시 메모리 등이 들어가 있는 임베디드 시스템이다.

Q)

가?

A)

•

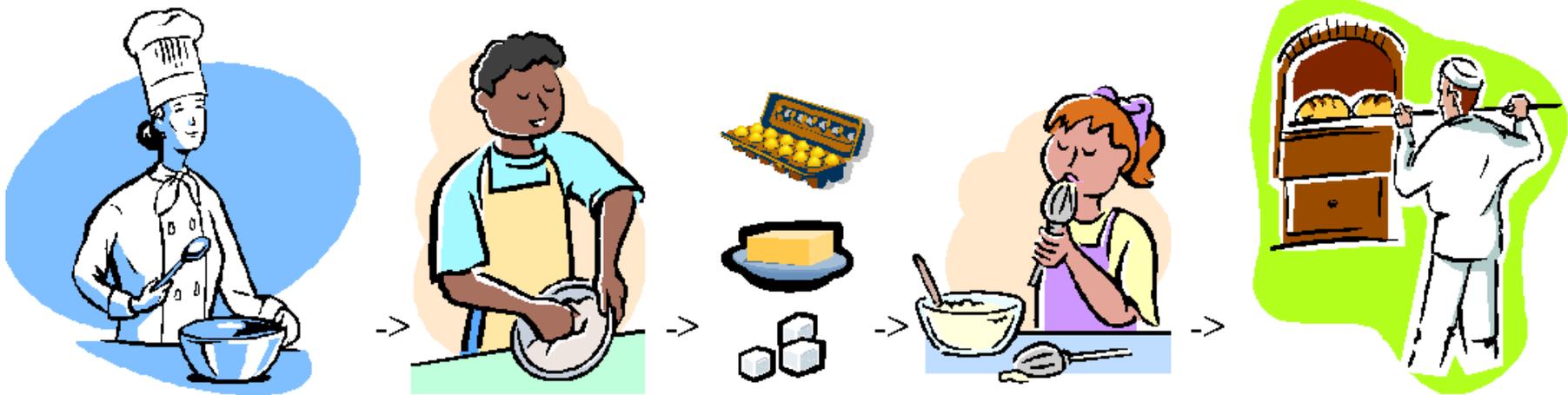
•

(algorithm):  
( )

가 가



가 , 가  
170~180



# 1 10

① 1부터 10까지의 숫자를 직접 하나씩 더한다.

$$I \quad 1 + 2 + 3 + \dots + 10 = 55$$

② 두수의 합이 10이 되도록 숫자들을 그룹핑하여 그룹의 개수에 10을 곱하고 남은 숫자 5를 더한다.

$$(0 + 10) = 10$$

$$(1 + 9) = 10$$

$$(2 + 8) = 10$$

$$(3 + 7) = 10$$

$$(4 + 6) = 10$$

5

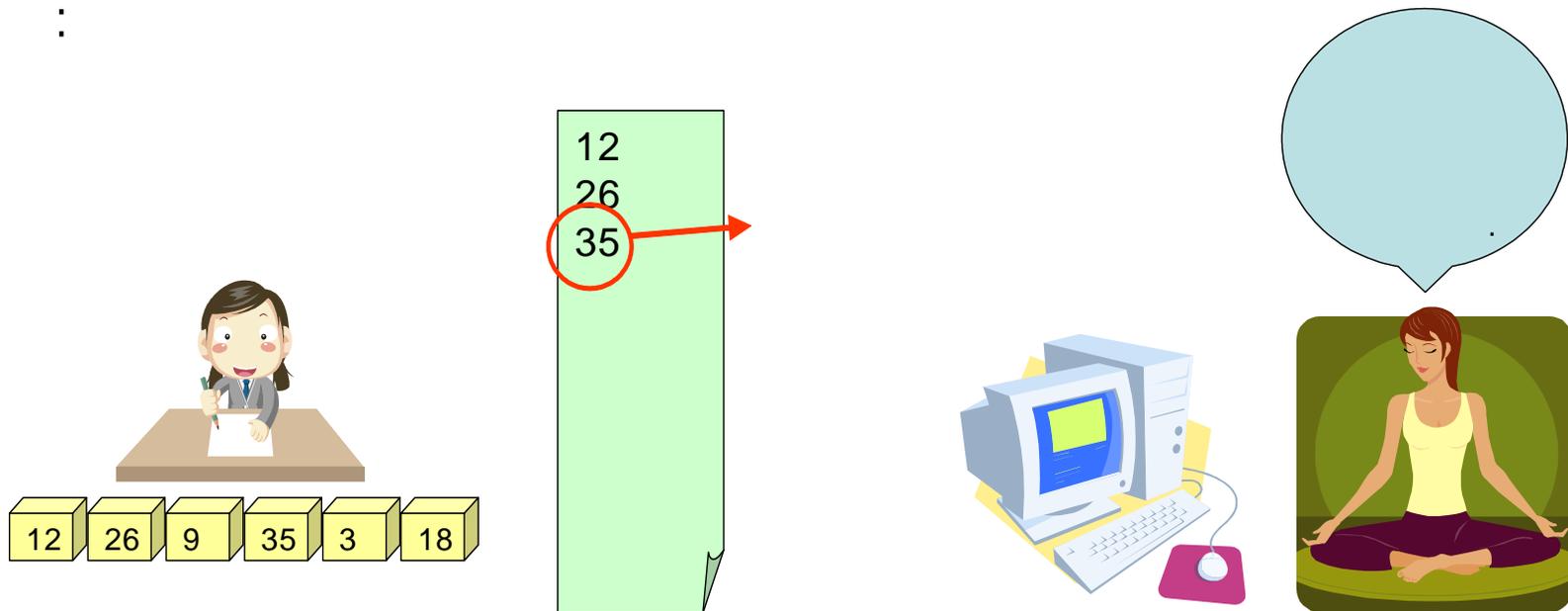
$$10 * 5 = 50 \quad + \quad 5 \quad = \quad 55$$

③ 공식을 이용하여 계산할 수도 있다.

$$10 * (1 + 10) / 2 = 55$$

- 1.
2. (flowchart)
3. (pseudo-code)

• :

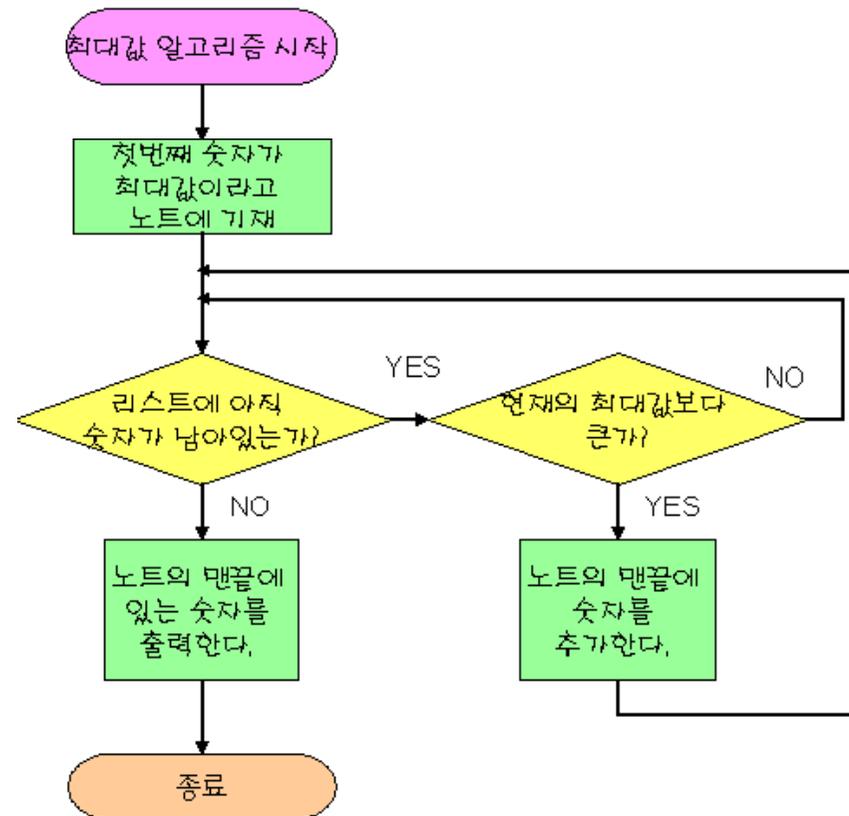


- (natural language)

- .

1.	가	가	가	.
2.	.			
3.	.		가	가

가



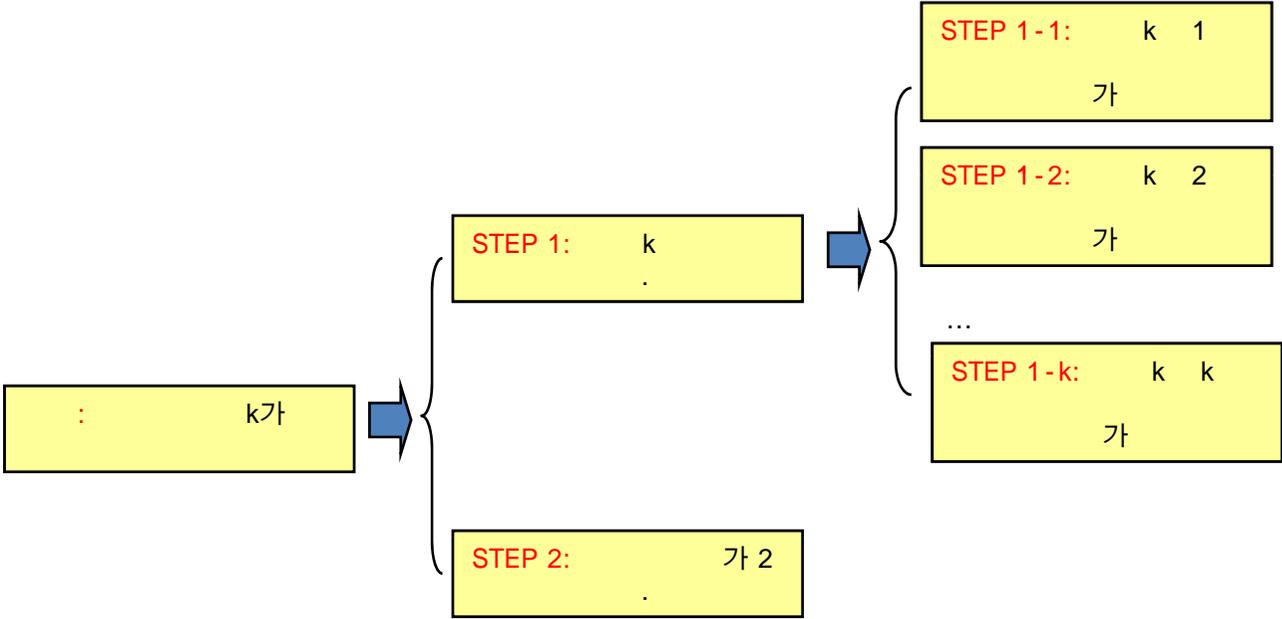
•

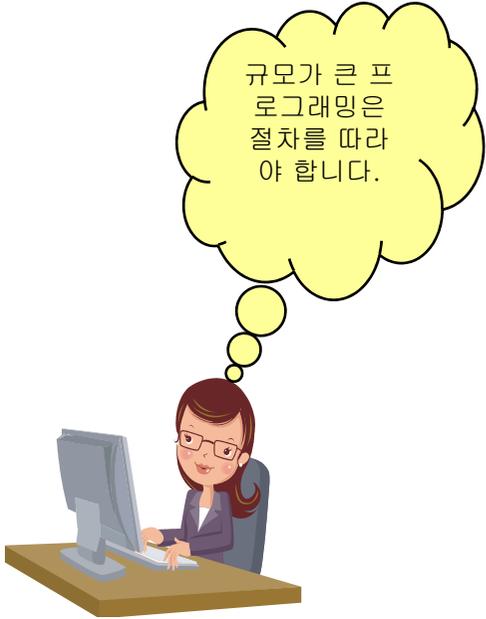
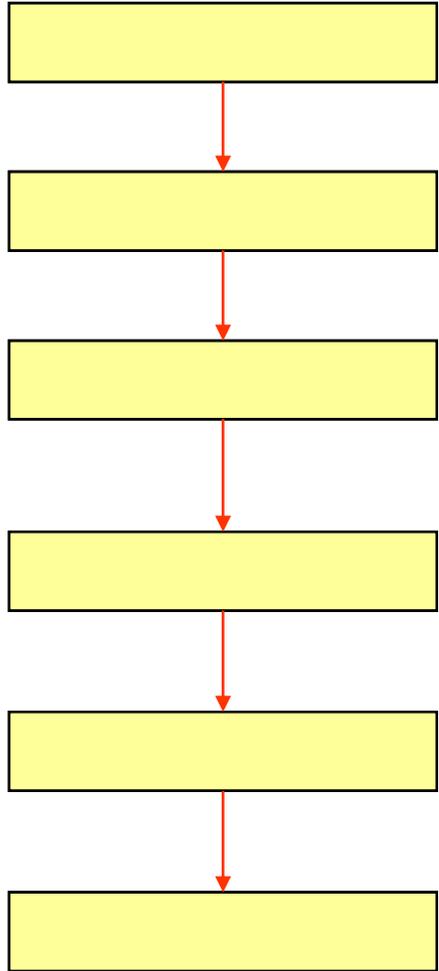
```
GetLargest
:
:
:
가
L.

largest = L[0]
for each n in L do
  if n > largest then
    largest = n
return largest
```

가 .

( ) k가 (prime) 가 가 .





•

• ( ) 3

—

or

—

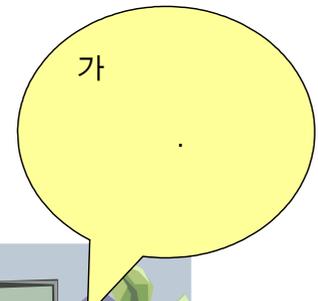
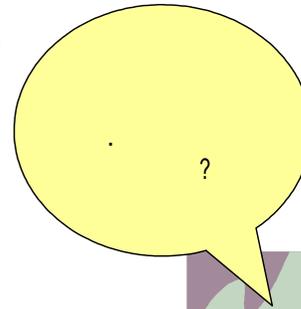
가 ?

•

:

가

,



- 
- 
- 
- 
- 



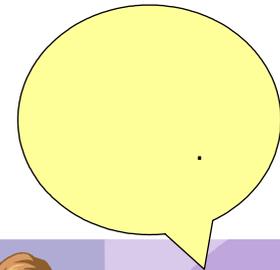
- 
- 
- 
- 
- 
- (Q)

가

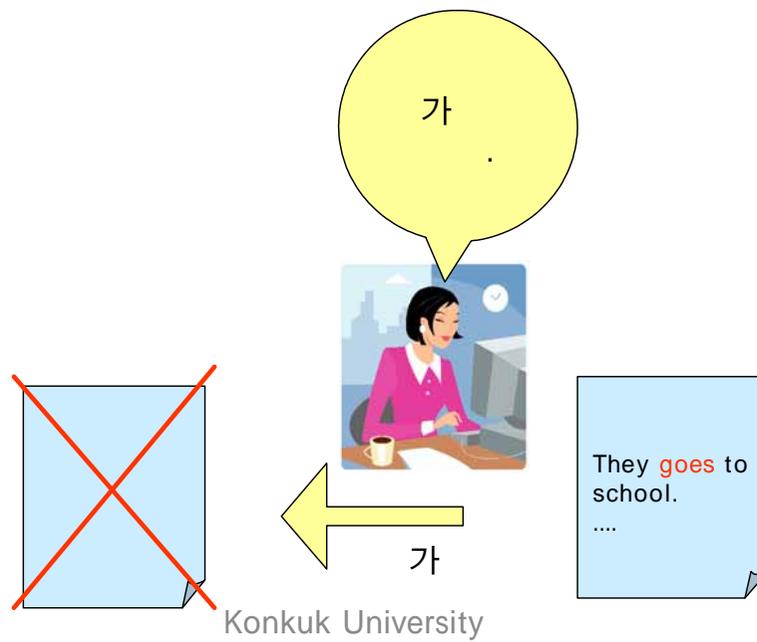
*(source program)*

?

(A) 알고리즘 개발이 더 창의적인 작업이고 더 어렵다



- 
- 가
- :
- ( ) He go to school;
- 가



- 

- 

*(library):*

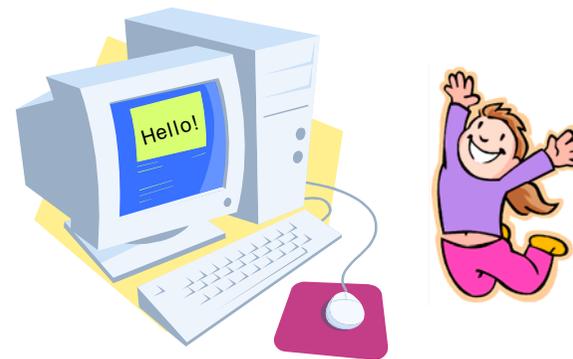
- - ( )

,

,

*(linker)*

.



- : .exe
- (run time error):

- 0

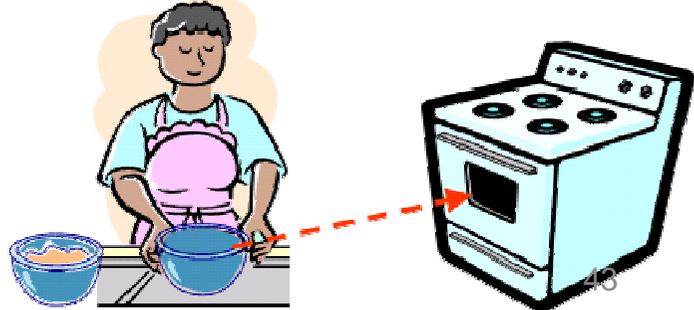
-

- (logical error):

- ( )

1	2		
1	가	,	
2		30	350

실수로 빈그릇  
을 오븐에 넣  
는다면 논리적  
인 오류입니다.





## 디버깅의 유래

1945년 마크 II라는 컴퓨터가 날아든 나방 때문에 고장을 일으켰고 이것을 "컴퓨터 버그(bug; 벌레)"라고 불렀다. 호퍼라는 사람이 나방을 채집해 기록에 남기고 이를 "디버깅(debugging)" 작업이라고 보고하였다. 이때부터 오류를 수정하는 작업을 디버깅이라고 부르기 시작하였다.

9/2  
9/9

0800 Antan started  
1000 " stopped - antan ✓ { 1.2700 9.030 117 025  
13'00 (032) HP-11C 2.130476495 9.017 846 995 count  
022 PRO = 2.130476495 4.615925019 (-2)  
count 2.130476495

Relays 6-2 in 033 failed speed speed test  
in relay 10.00 sec.

1100 Started Cosine Tape (Sine check)  
1525 Started Multi Adder Test.

1545 Relay #70 Panel F  
(moth) in relay.

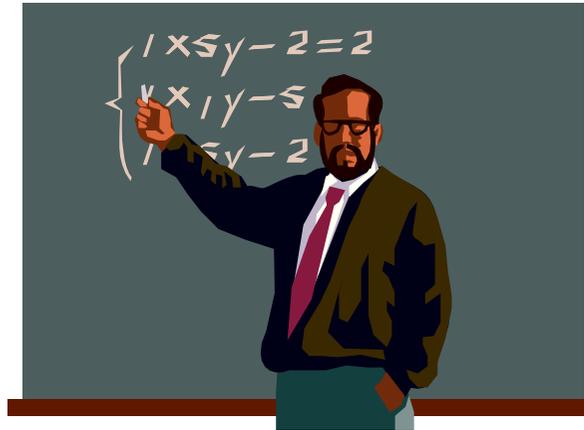
First actual case of bug being found.  
1630 antan started.  
1700 closed down.

Relay #70  
2145  
Relay #33

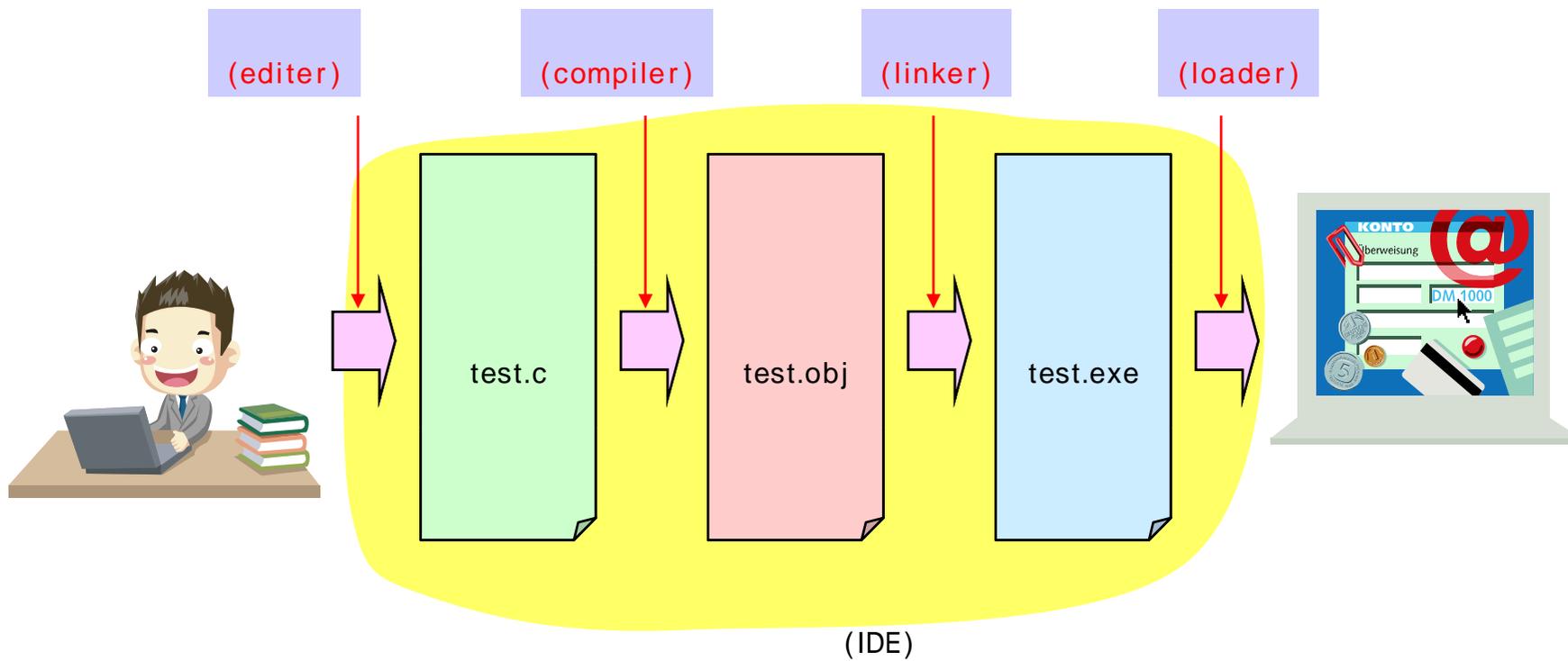
- 가  
 - 가  
 - 가 가
- 50%



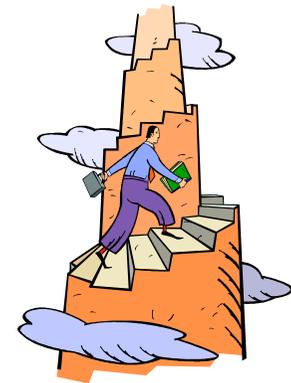
# Q & A

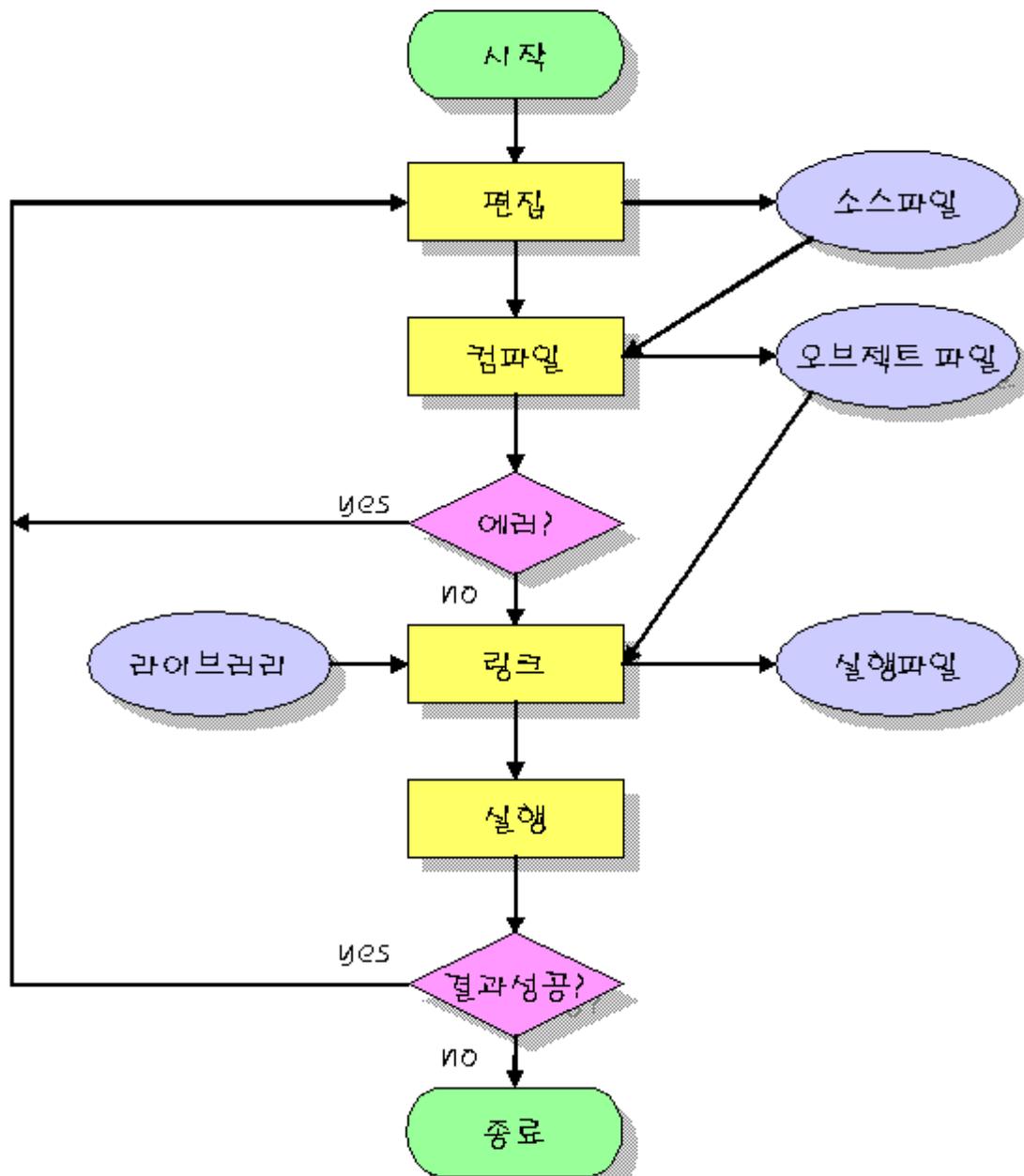


2



- (edit)
  - 
  - (source file): 가
    - ( ) test.c
  
- (compile)
  - ->
  - (object file) :
    - ( ) test.obj
  
- (link)
  - 
  - (executable file): 가
    - ( ) test.exe





# Q & A

- (Q)

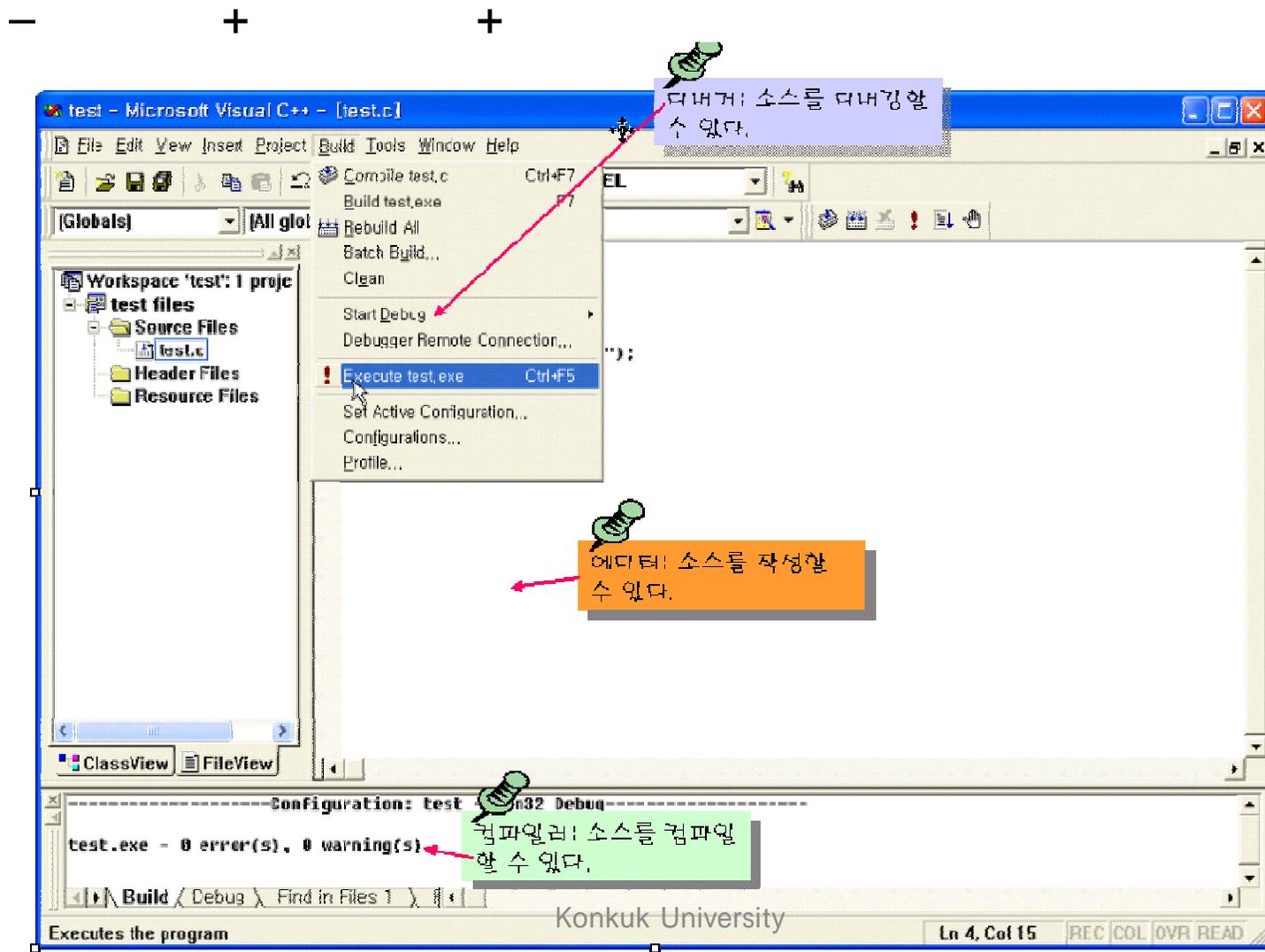
?



- (A)

가  
. Visual C++

- (IDE: Integrated Development Environment):

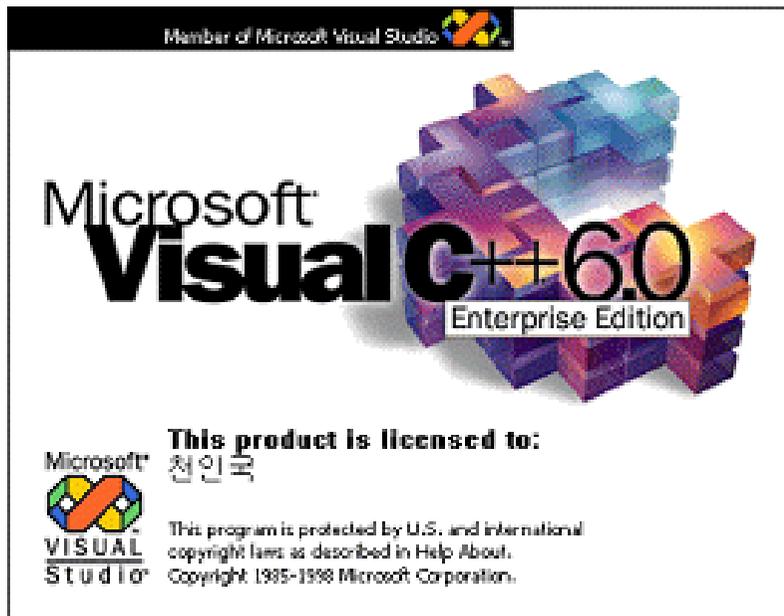


- C++(Visual C++)

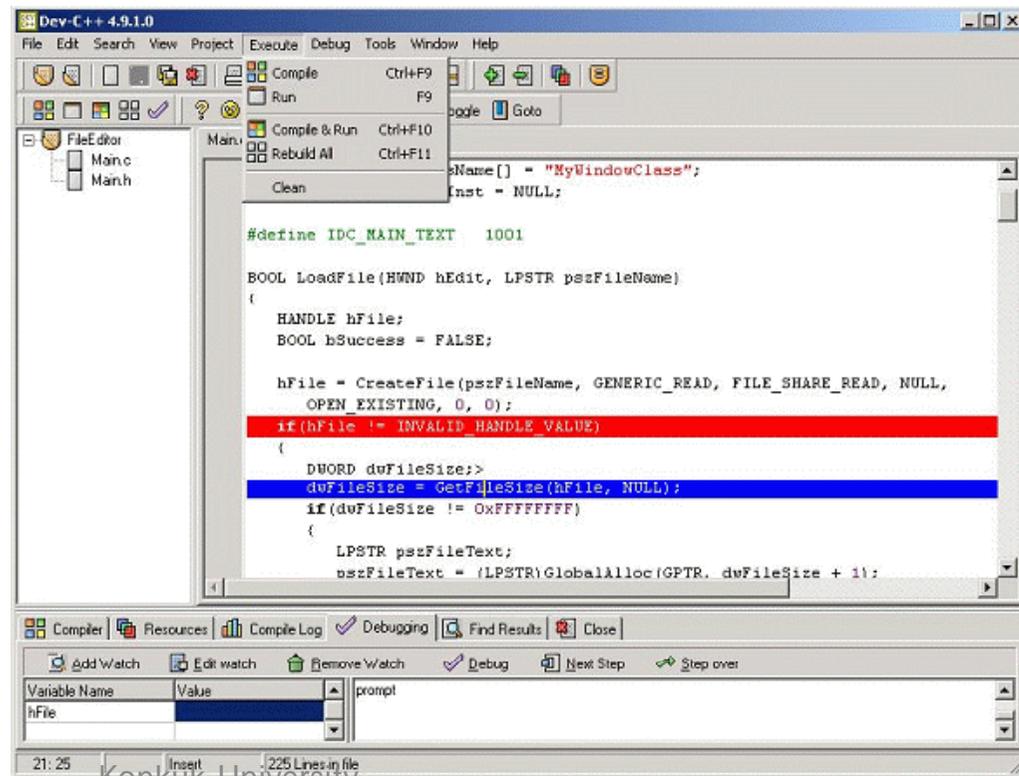
- 
- 
- 
- 

가 : 2005  
 가 : \_\_\_\_\_ 6.0

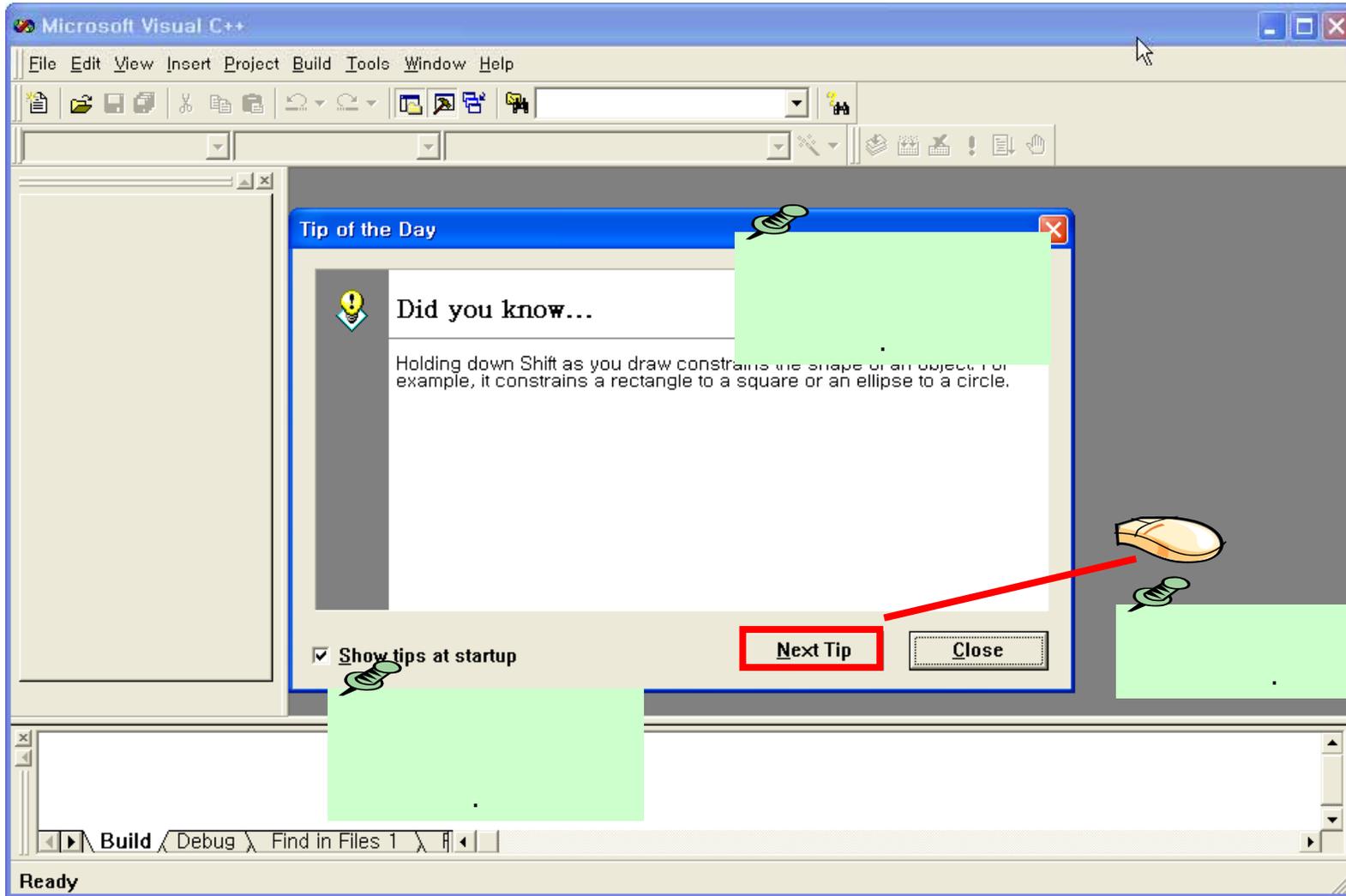
가



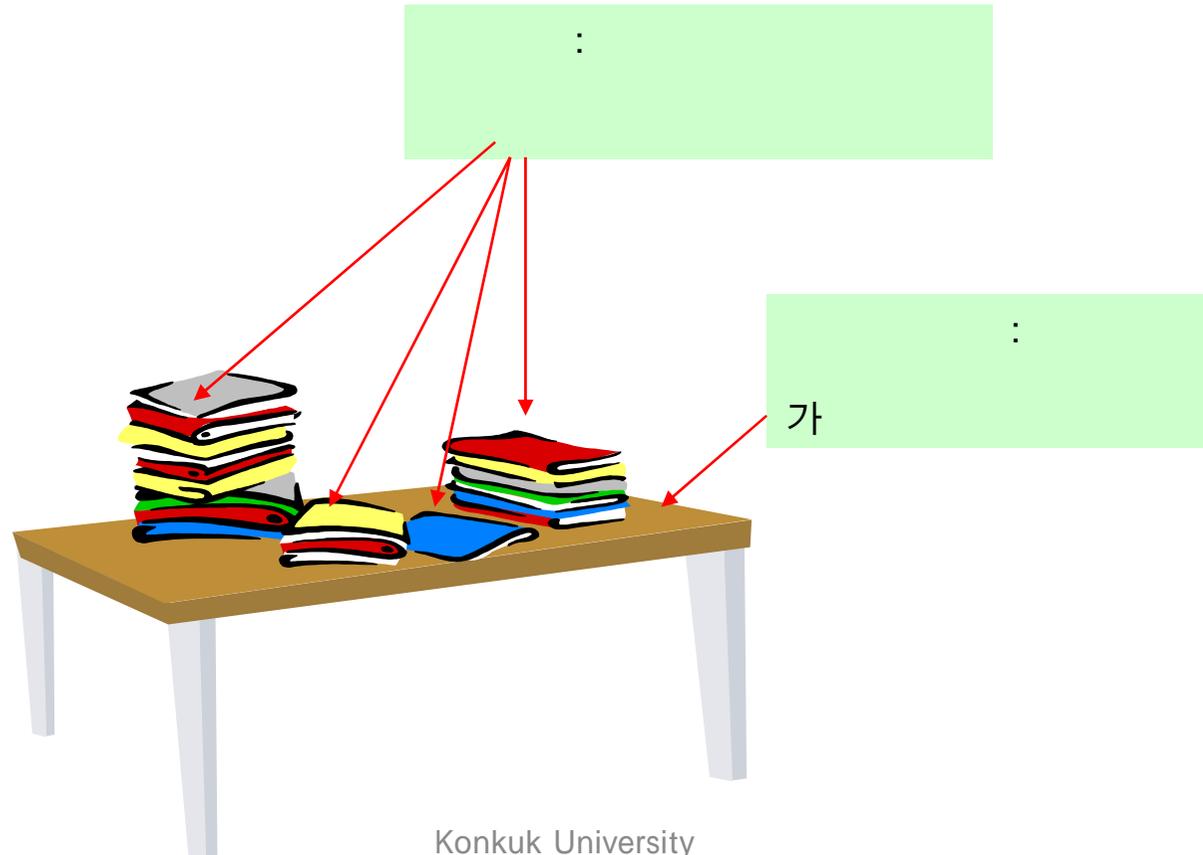
- Dev-C++
  - 
  - C/C++
  - GCC
  -



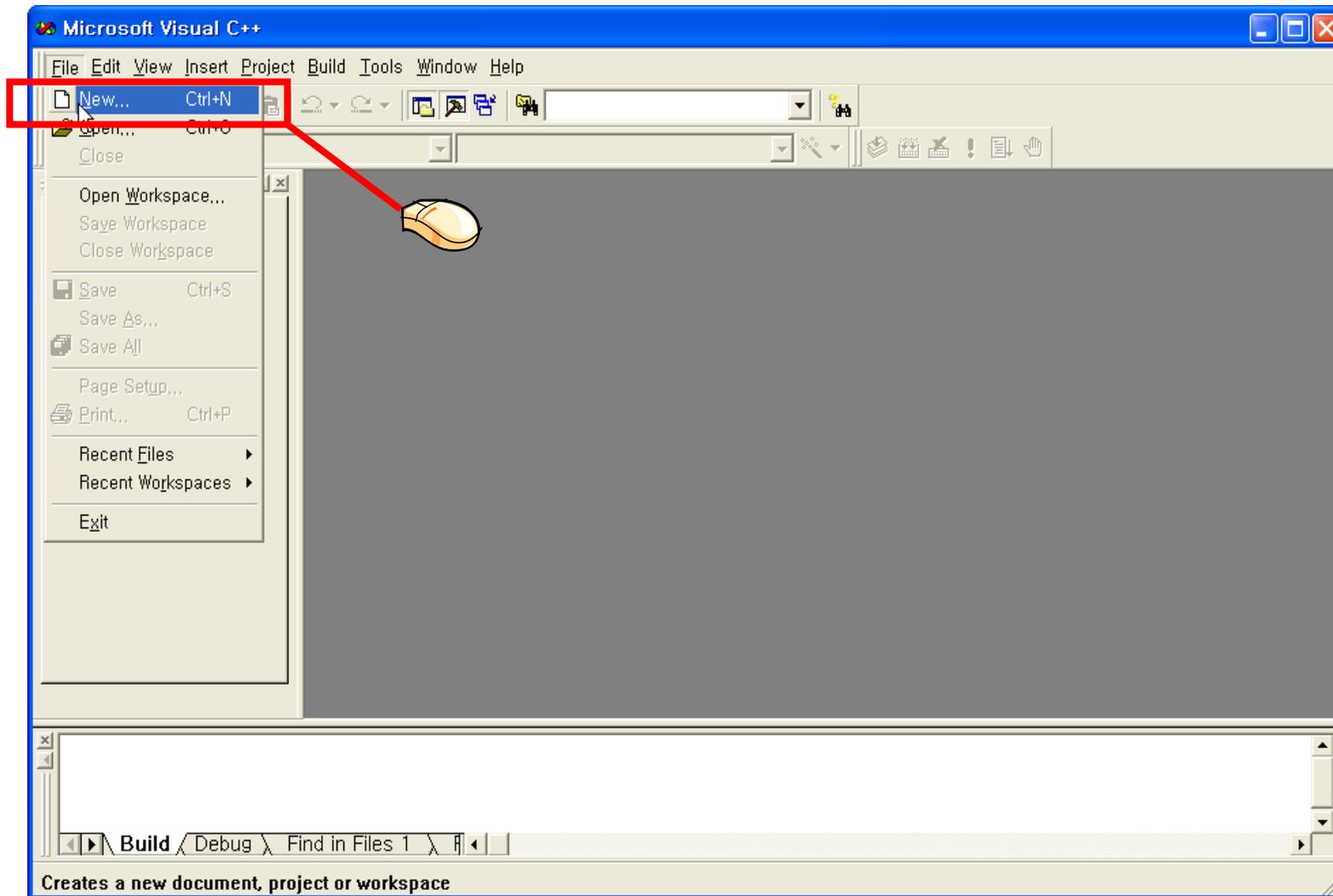
# C++

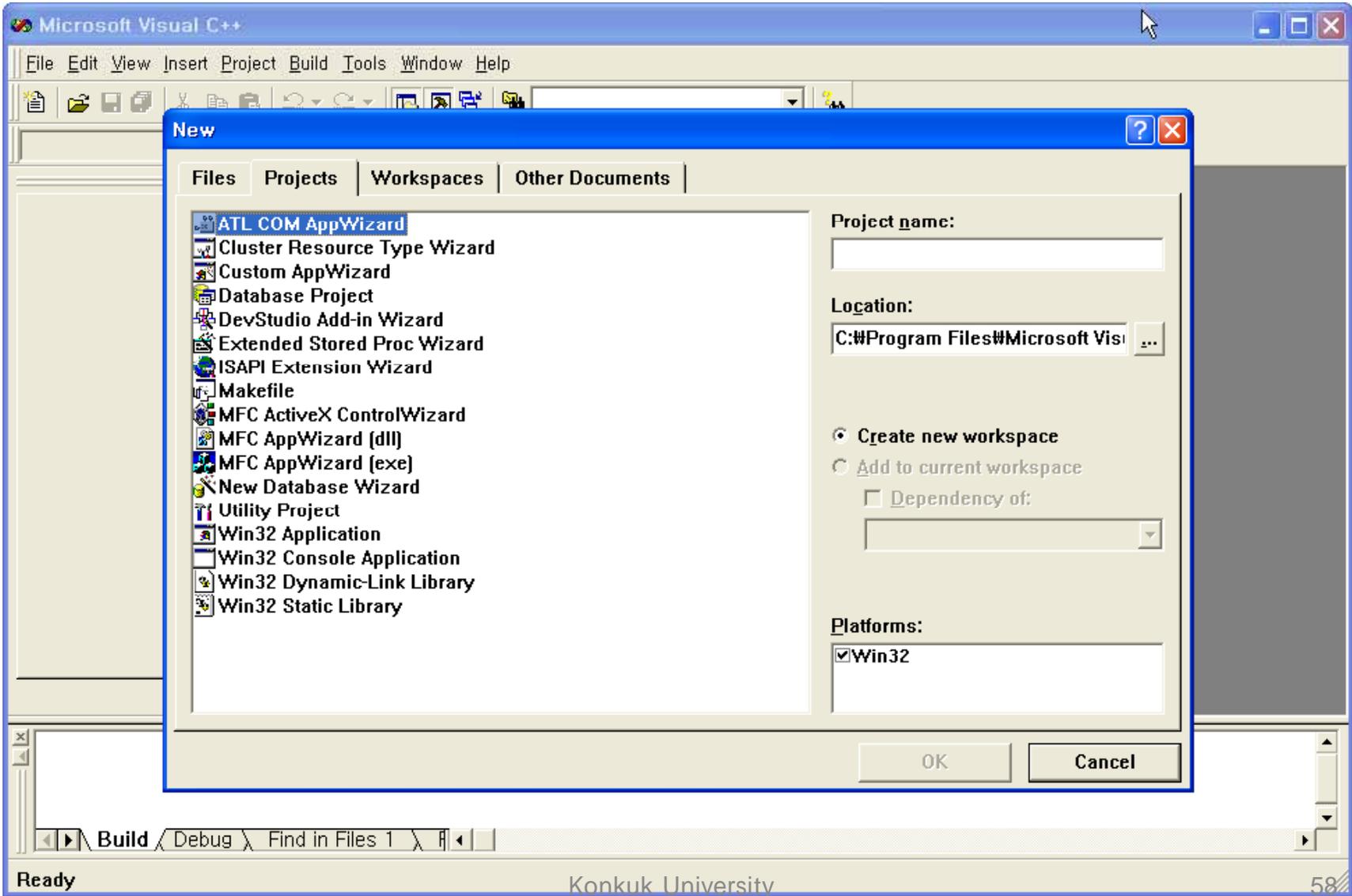


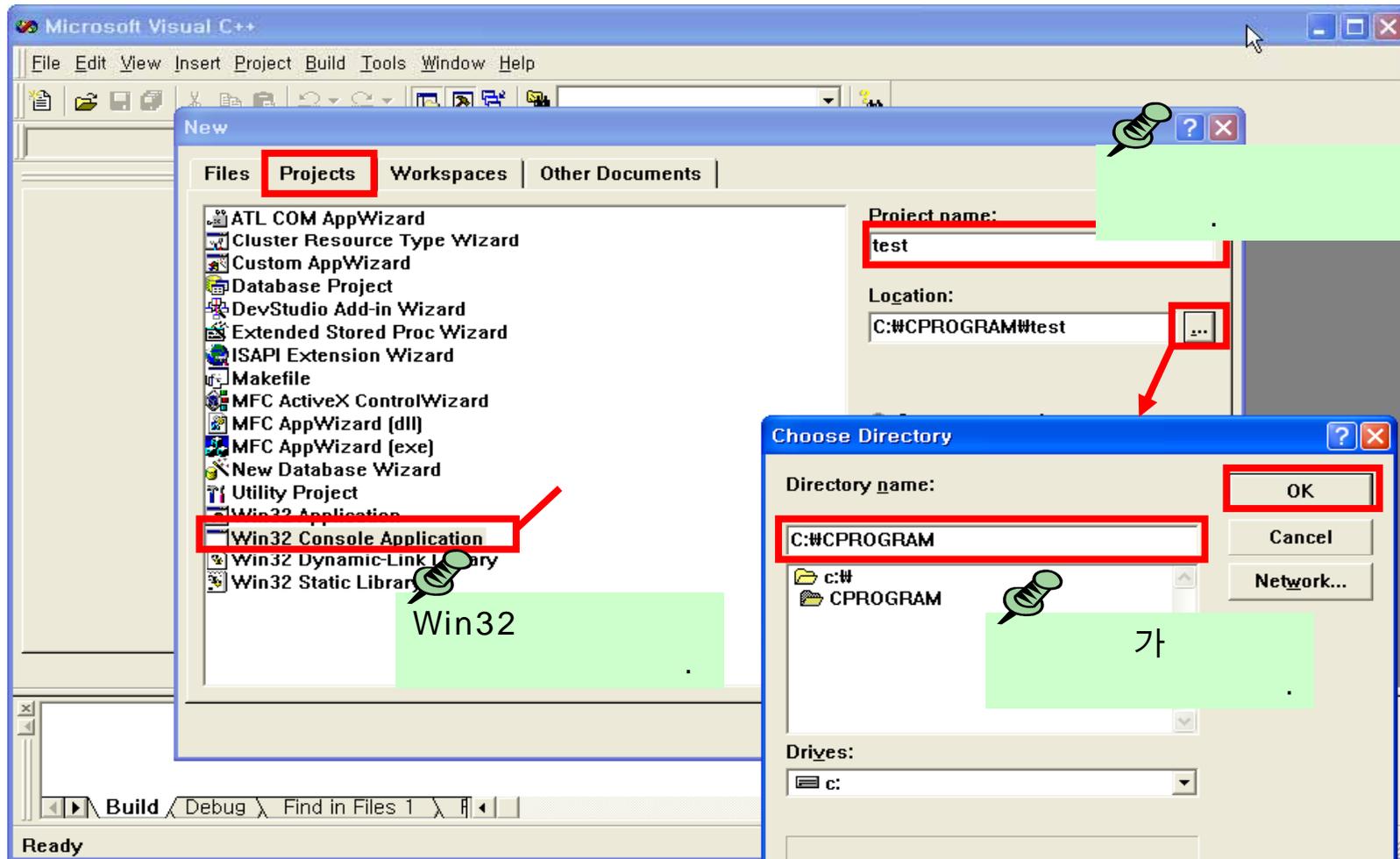
- (workspace); , 가
- (project):

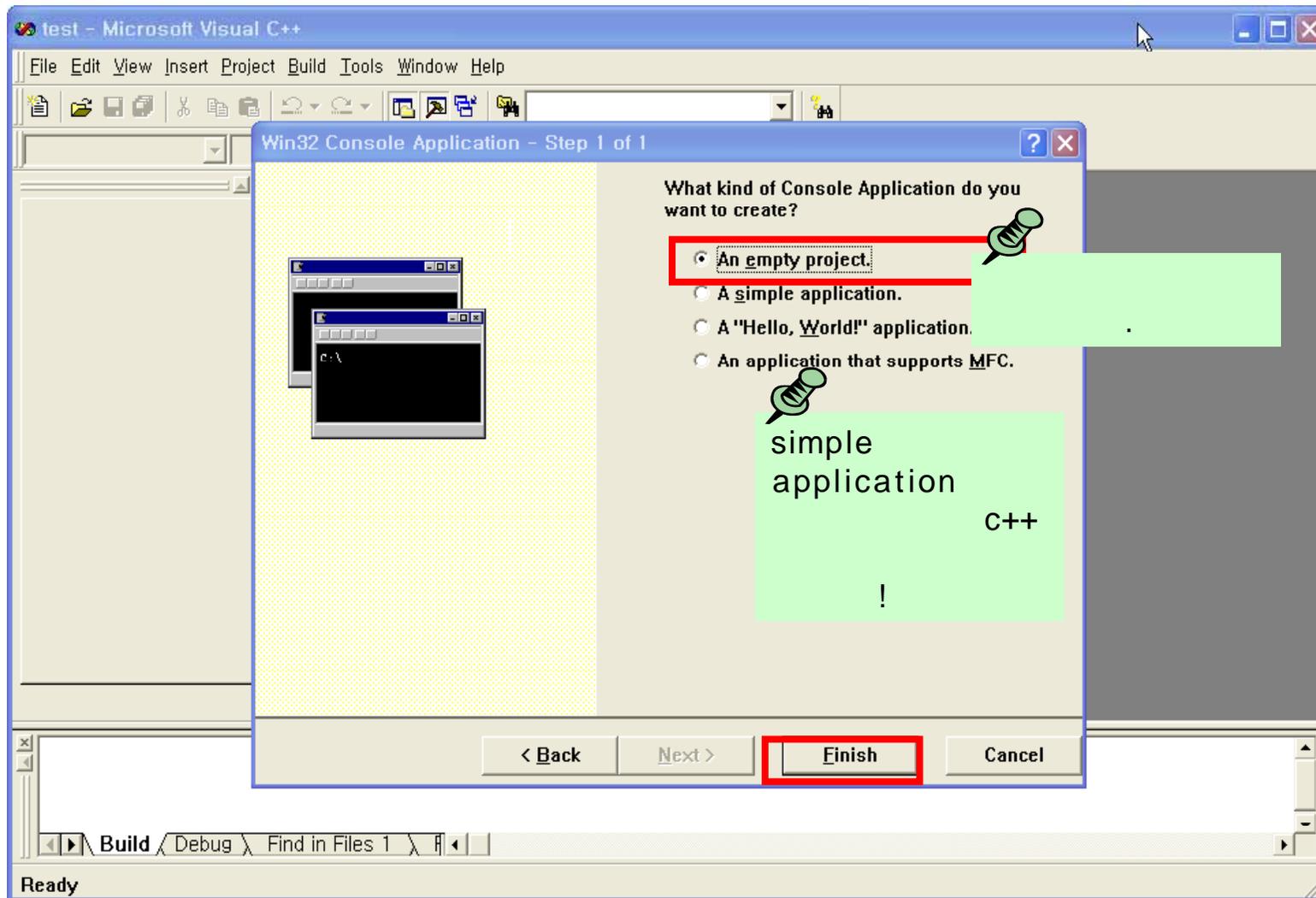


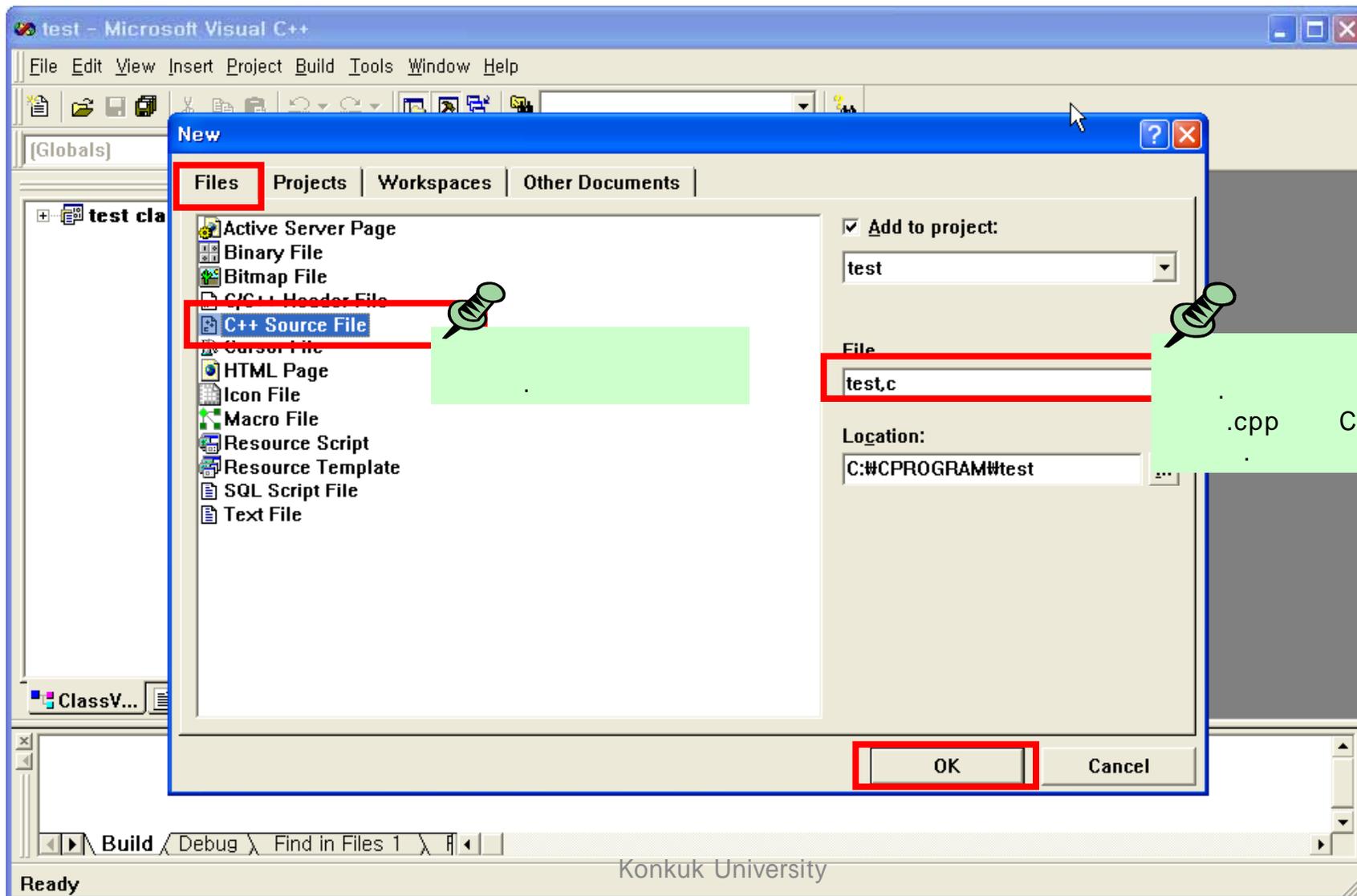
# C++

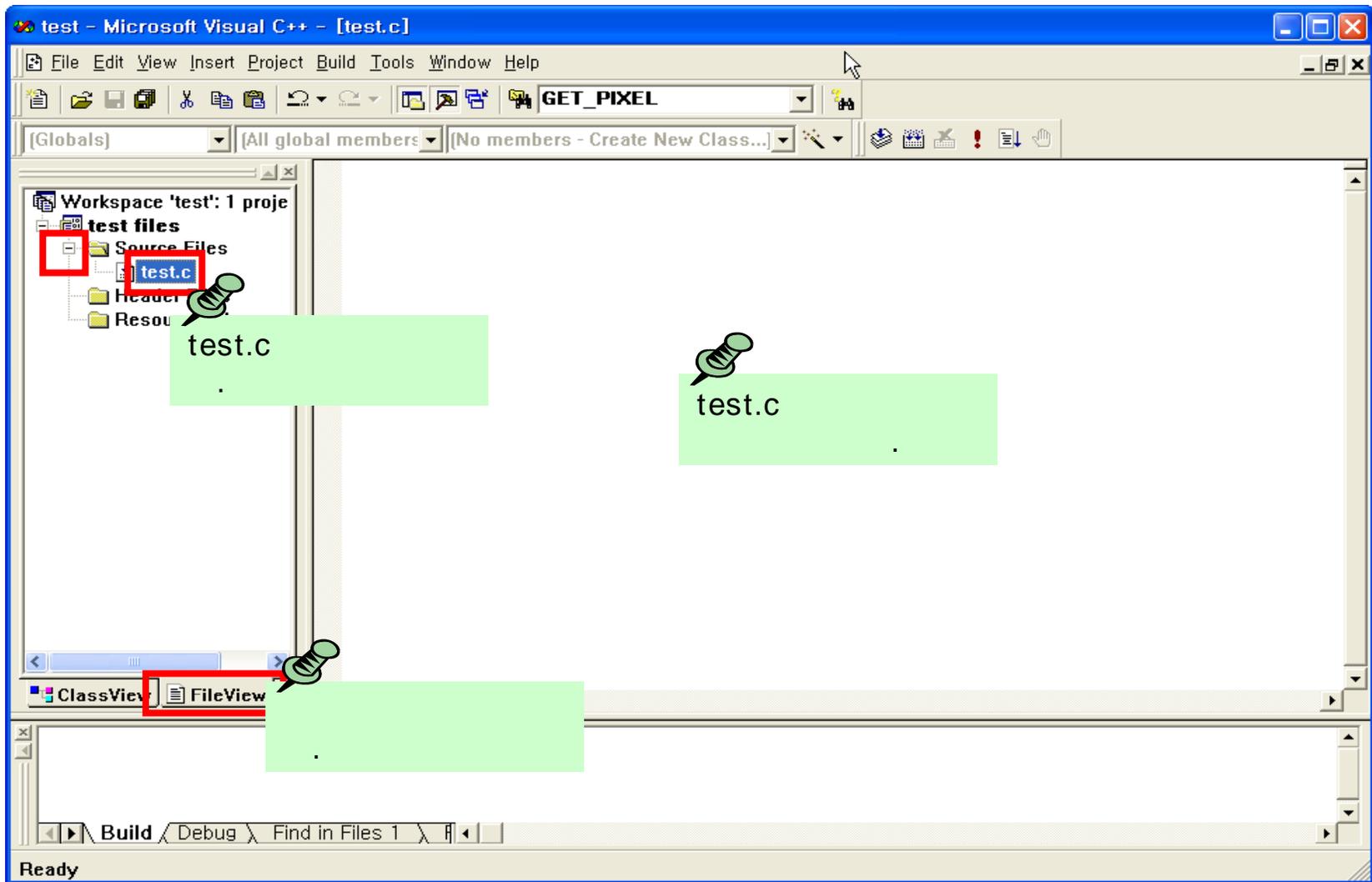




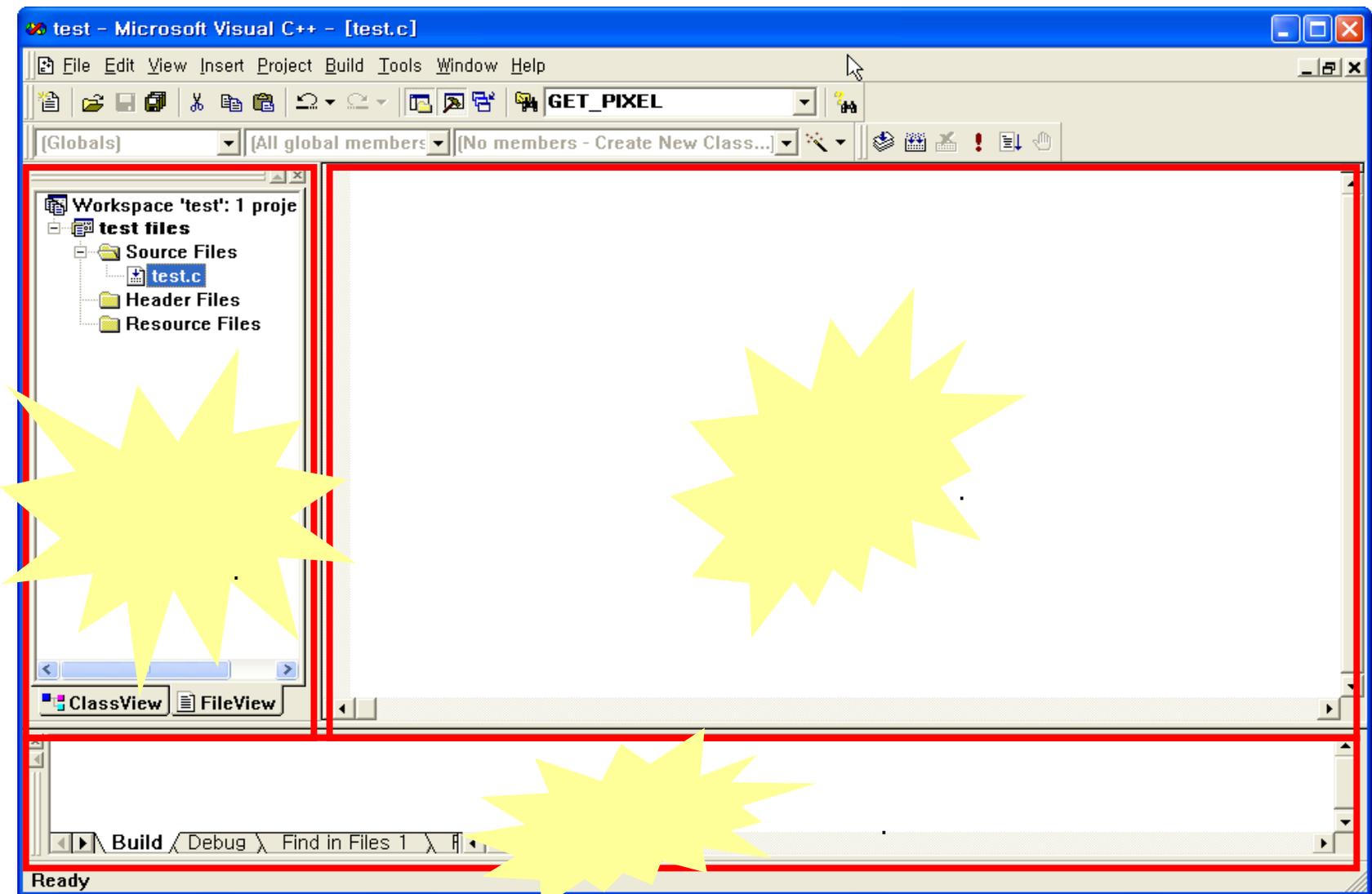


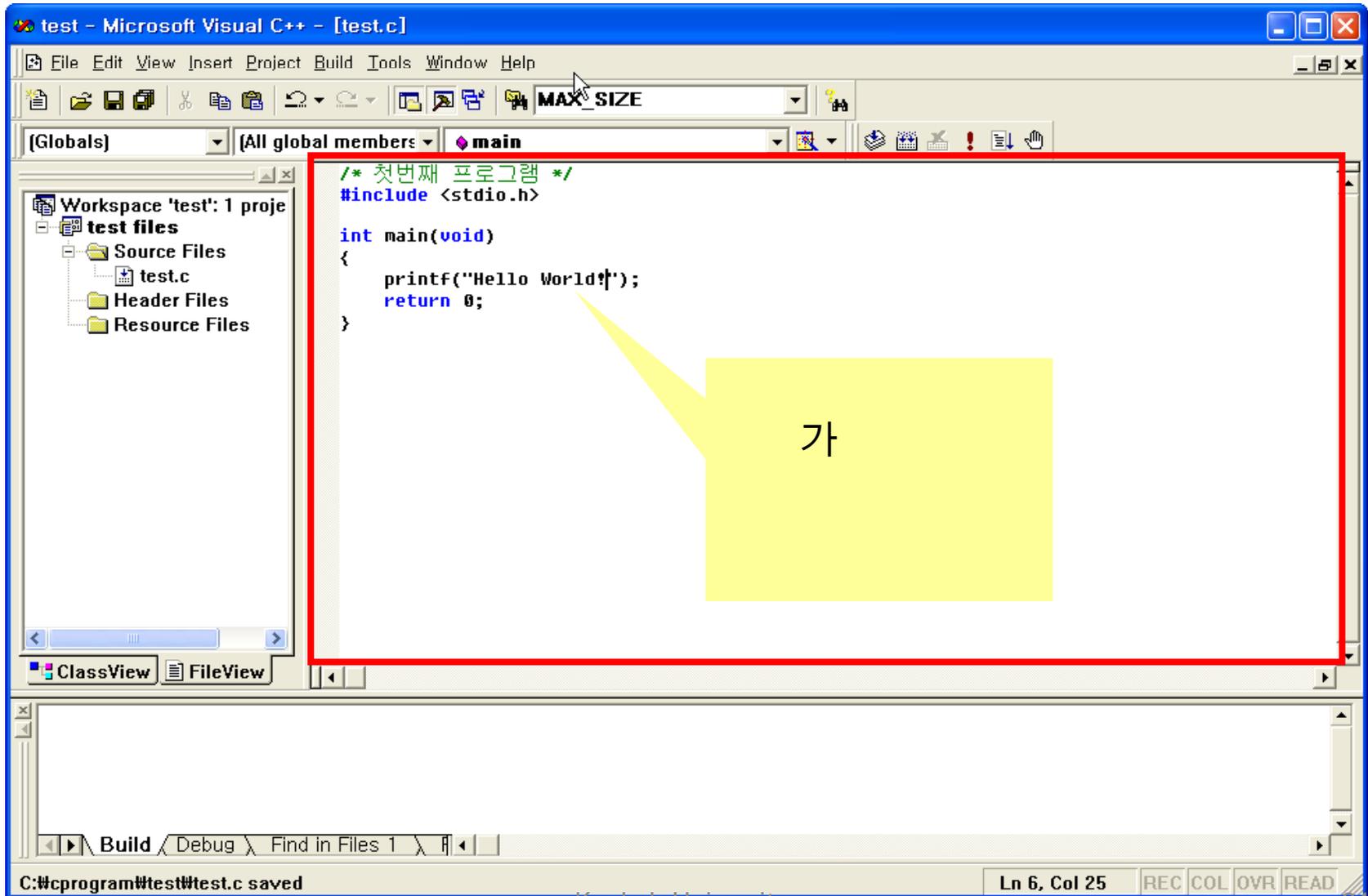






# C++





/\*와 \*/에서 /와 \*는 반드시 붙여서 쓰도록 한다.

include나 stdio와 같은 단어는 붙여서 쓴다.

int와 main은 별도의 단어로 구분하기 위하여 공백이 있어야 한다.

```
/* 첫째 프로그램 */
#include <stdio.h>

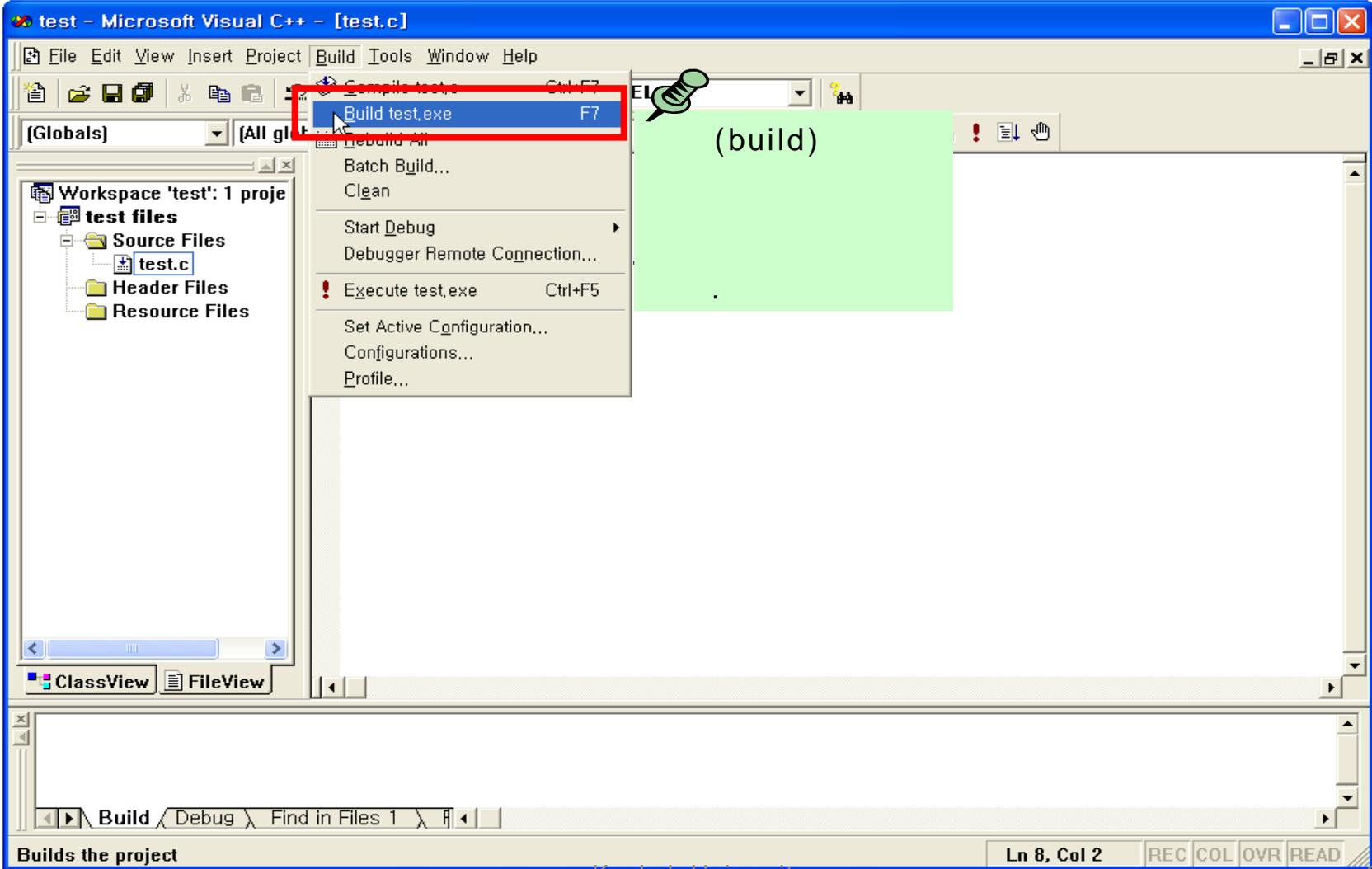
int main(void)
{
    printf("Hello World!");
    return 0;
}
```

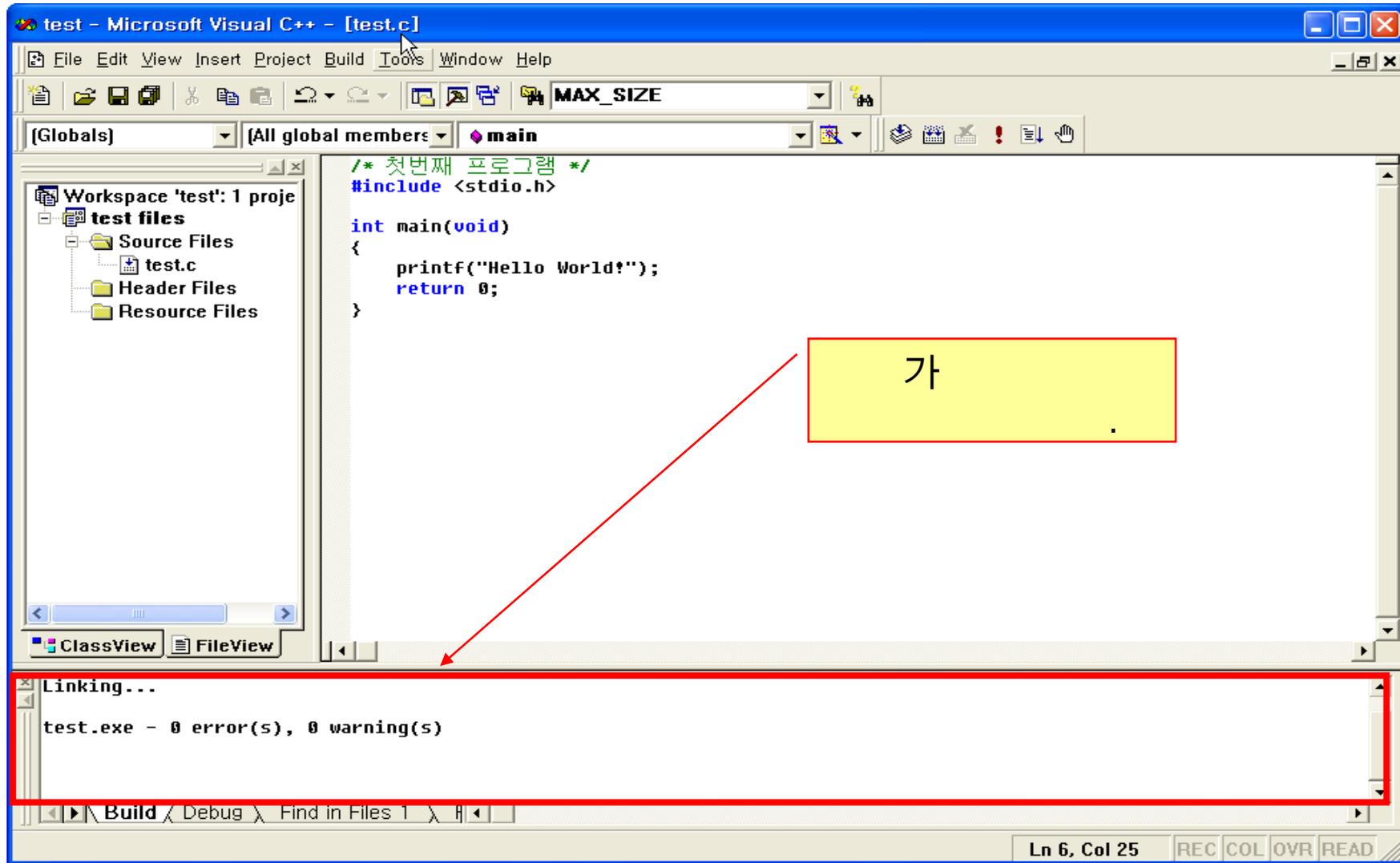
큰따옴표안의 문장들은 화면에 그대로 출력된다. 여기서 \와 n은 반드시 붙여야 한다.

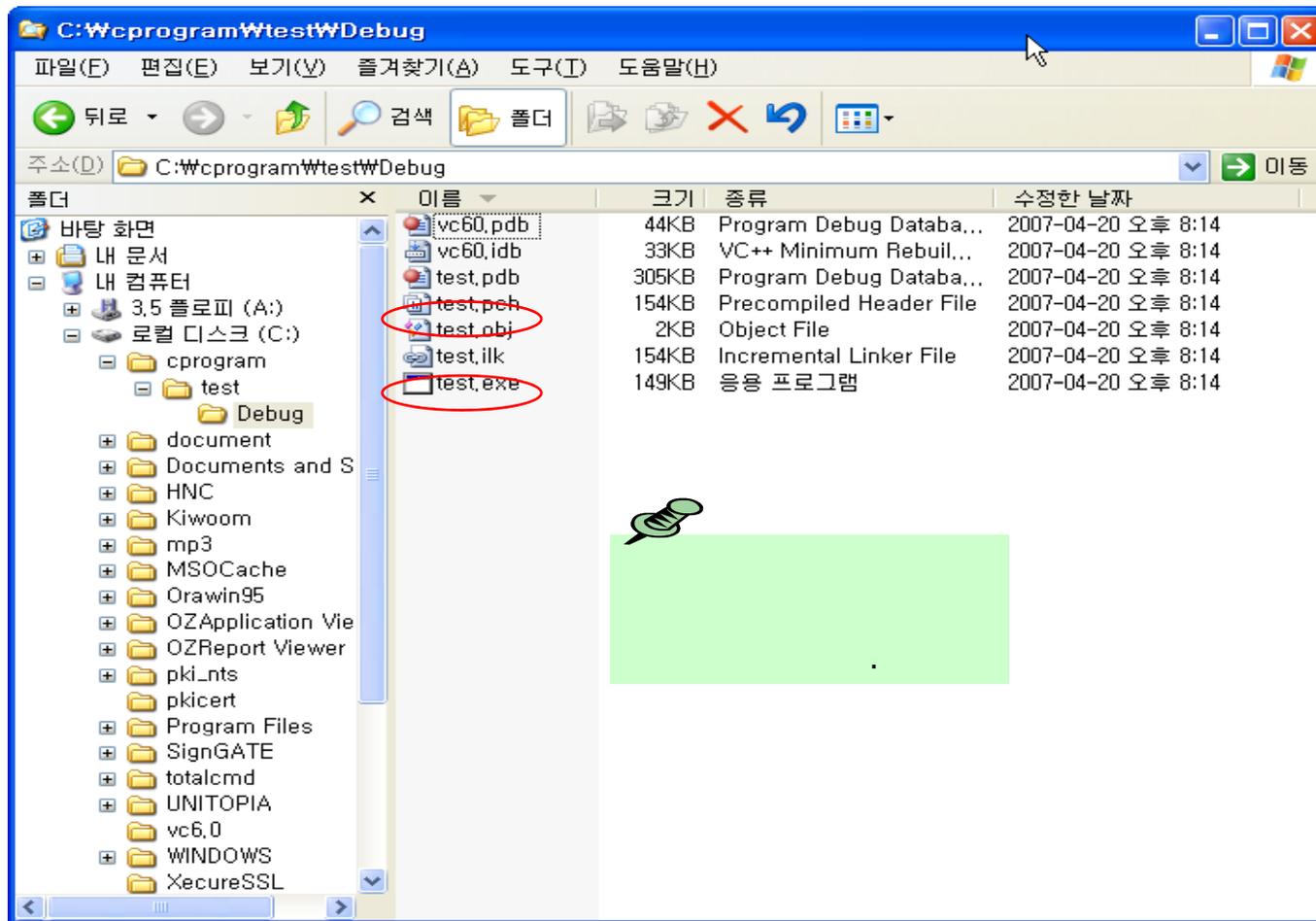
문장의 끝에는 ;을 잊지말자. ;와 :을 잘 구별한다.

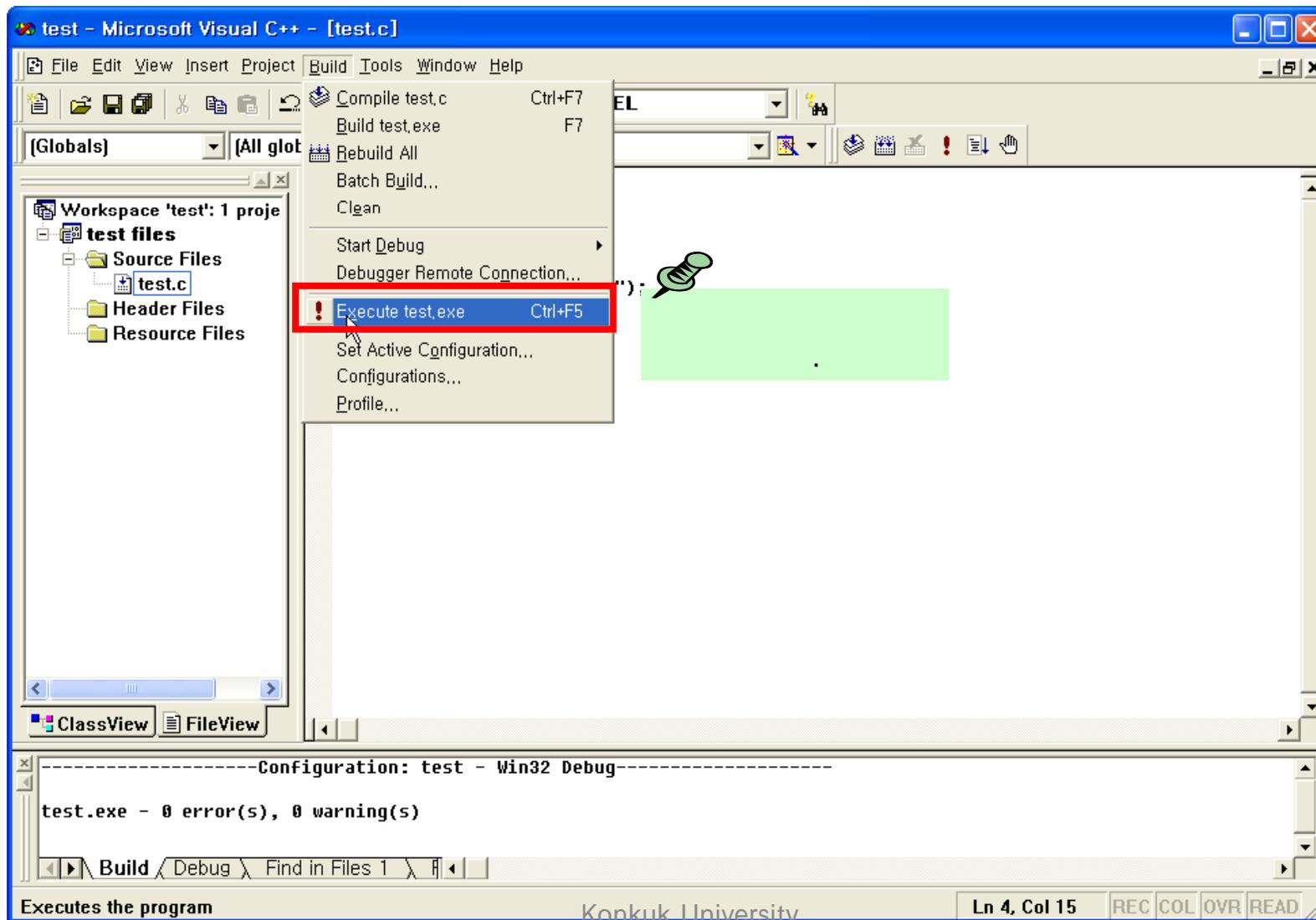
서로 대응되는 중괄호들은 같은 열에 놓는 편이 좋다.

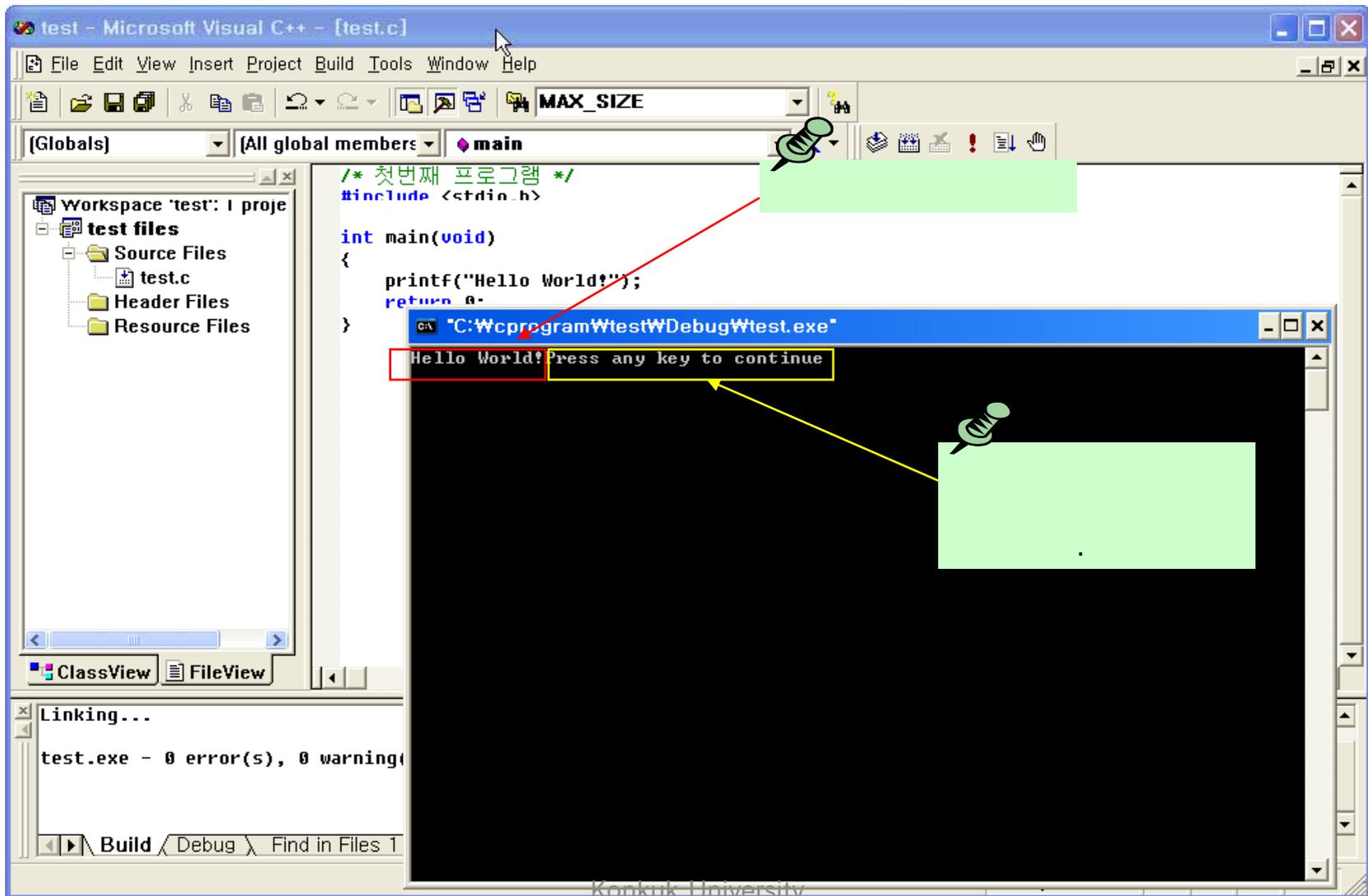
중괄호안에 들어가는 문장들은 일반적으로 들여쓰기를 한다. 탭키를 이용하거나 스페이스키를 이용한다. 비주얼 C++에는 자동적으로 들여쓰기를 해주는 기능이 있다.











```
/*          */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!");  
    return 0;  
}
```



- (comment):

```
/*          */  
  
int main(void) /*          */  
  
/*  
          */
```

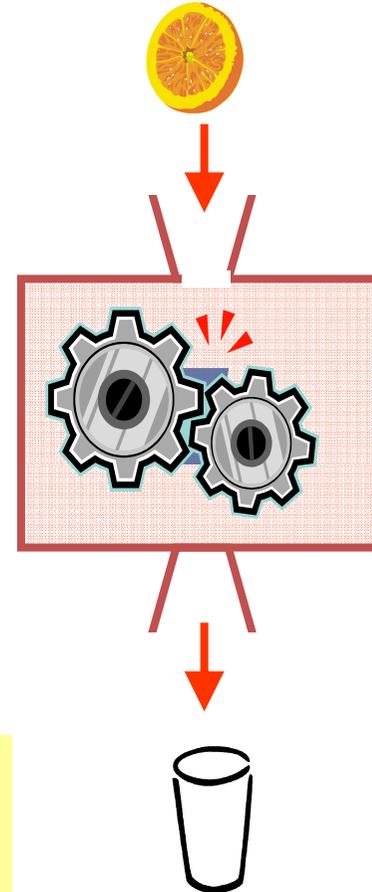


```
#include <stdio.h>
```

- #include
- (header file): 가 가
- stdio.h: standard input output header file
- !:

```
int main(void)
```

- (function):
- ( )  $y = x^2 + 1$
- =
- main() 가



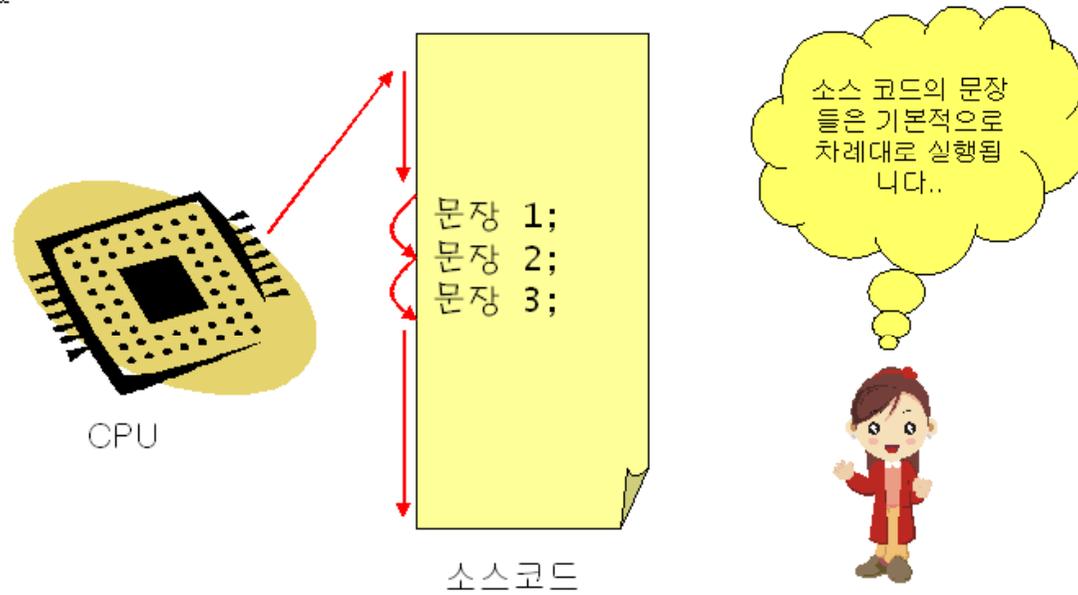
```
int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

함수의 출력 타입  
함수의 이름  
함수의 입력 타입

함수의 시작  
함수의 몸체  
함수의 시작

- 
- 

I



# printf()

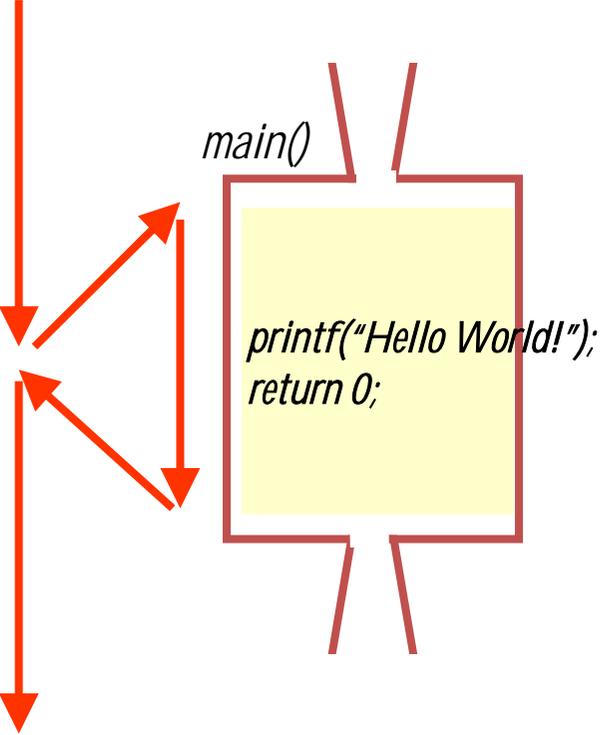
```
printf("Hello World!");
```

- printf() 가
- 



```
return 0;
```

- return



#1

가

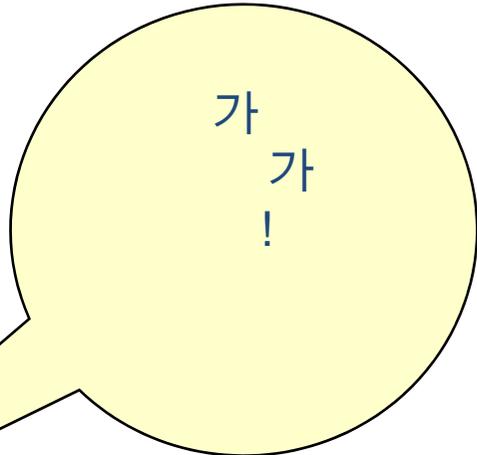


- 

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

int main(void)
{
    printf("Hello World!");
    printf("Kim ChulSoo");
    return 0;
}
```

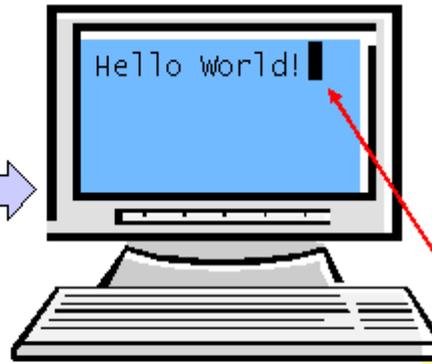
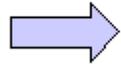
Hello World!Kim ChulSoo



\n

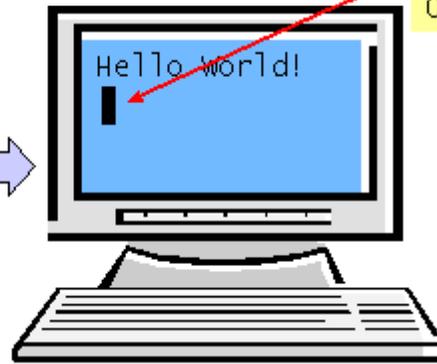
\n

`printf("Hello World!");`



현재 커서의 위치,  
다음 문자를 표시할 때는  
이곳부터 시작한다.

`printf("Hello World!\n");`



- 

가 가 .

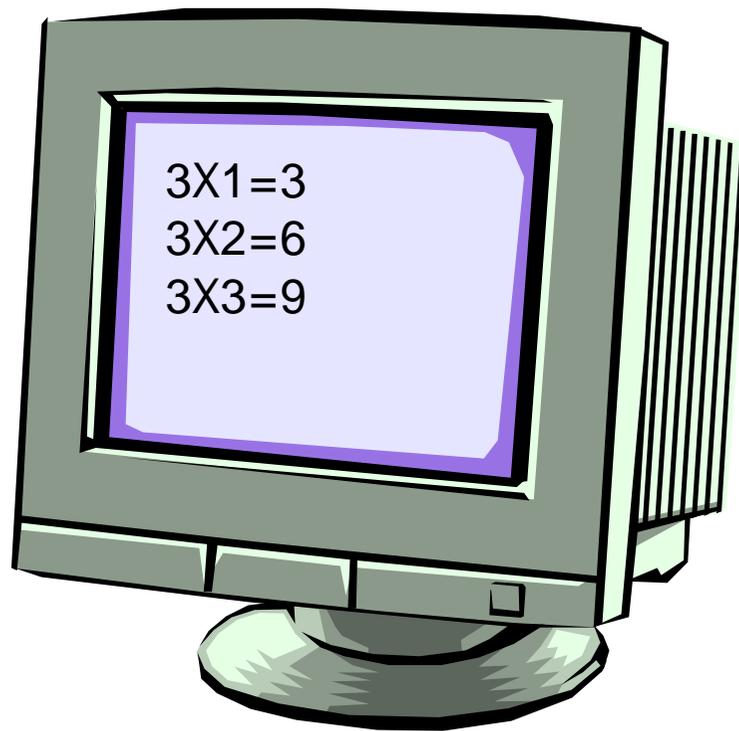
```
/*  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n");  
    printf("Kim ChulSoo");  
    return 0;  
}
```

```
Hello World!  
Kim ChulSoo
```



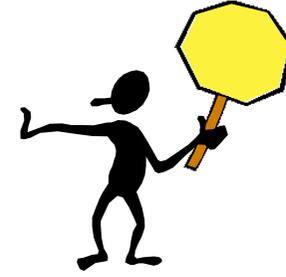
# #2

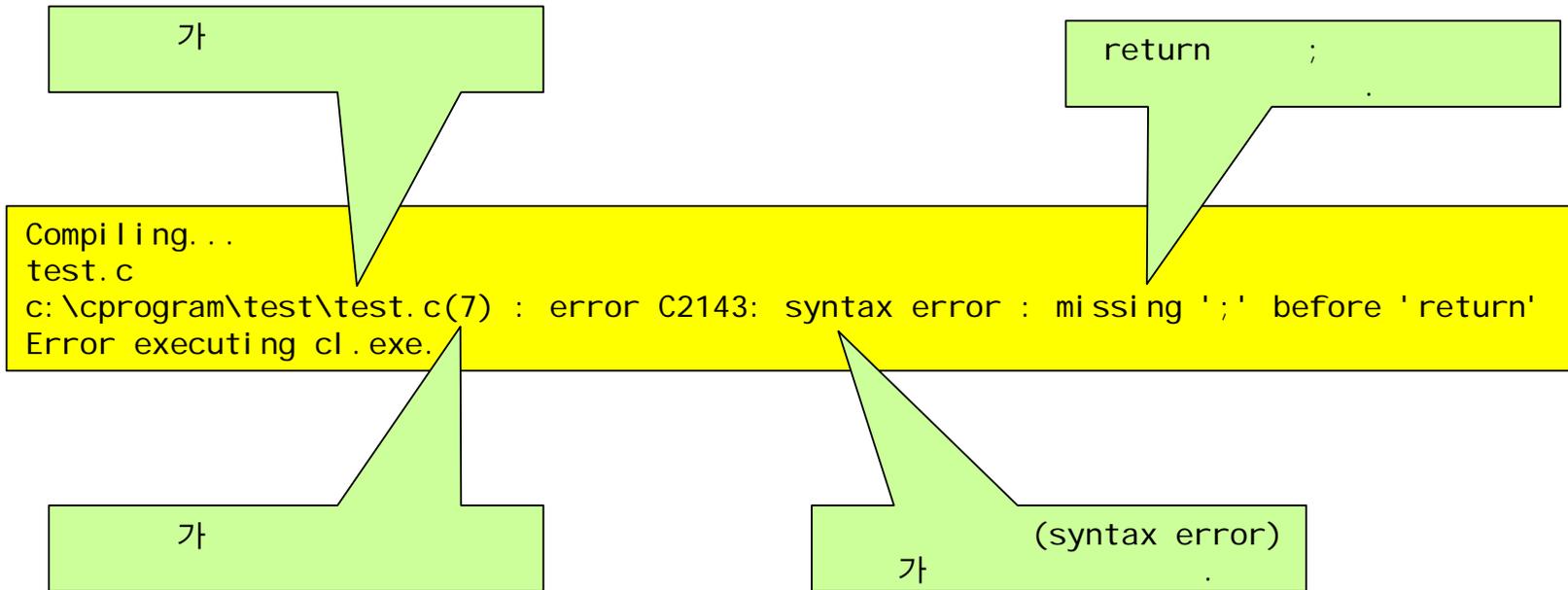
가

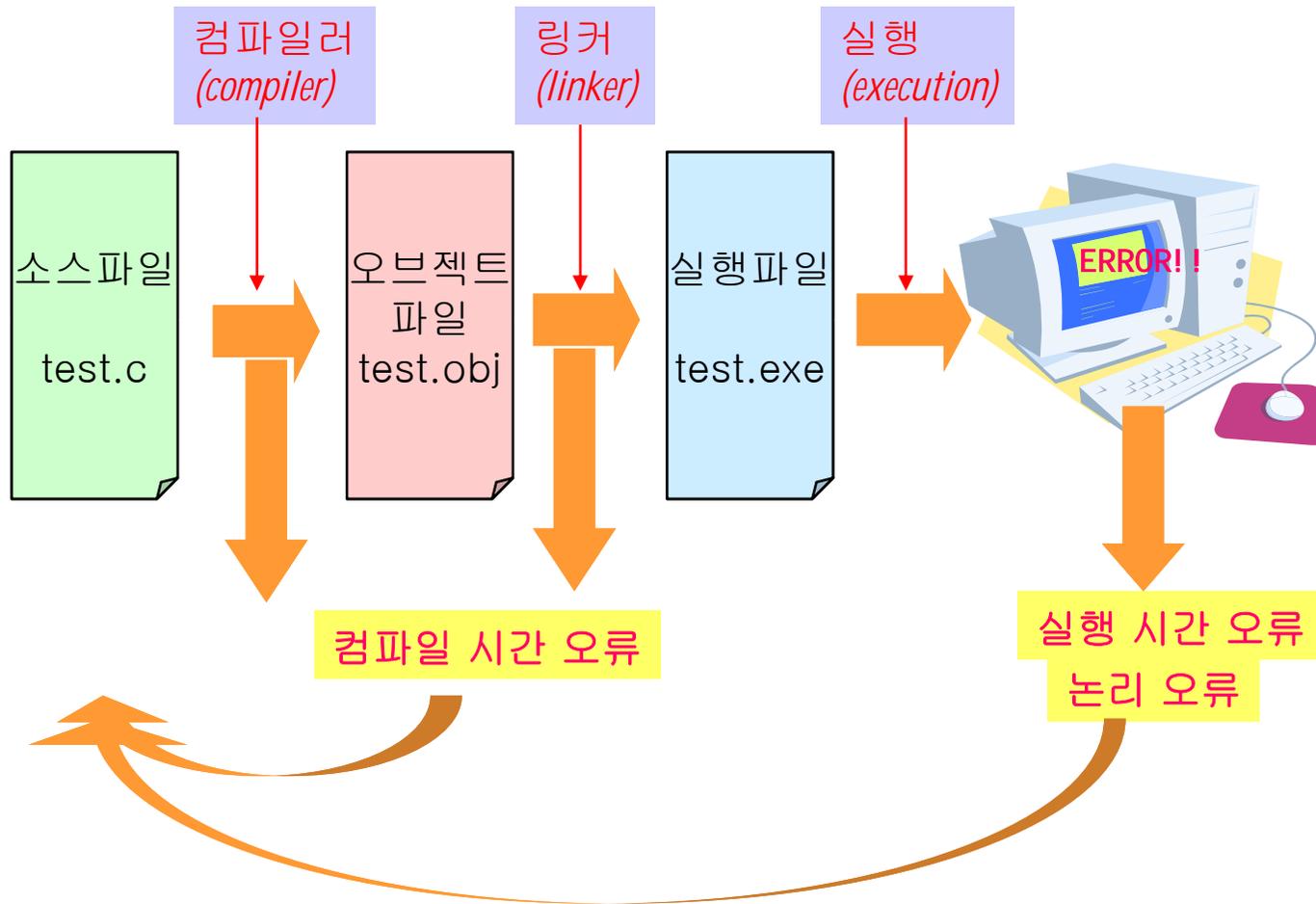


```
/*                                     */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("3 X 1 = 3\n");  
    printf("3 X 2 = 6\n");  
    printf("3 X 3 = 9\n");  
  
    return 0;  
}
```

- 가 .
- 
- (error):
- (warning):
- 
- :
- :
- : 0 가

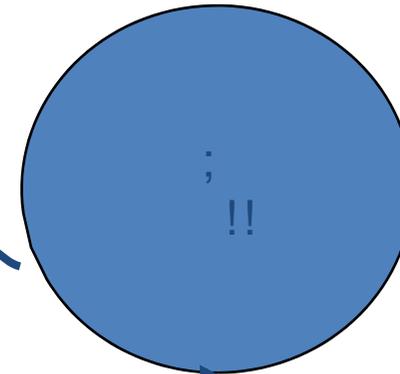






# #1

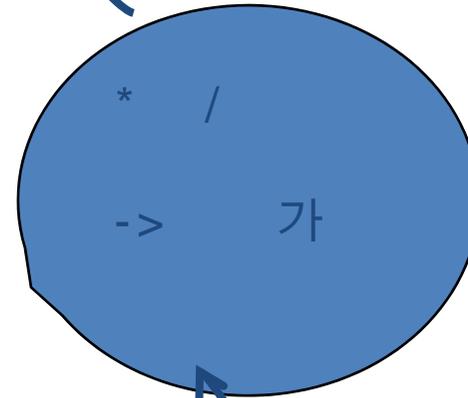
```
/*   가   */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n")  
    return 0;  
}
```



```
-----Configuration: test - Win32 Debug-----  
Compiling...  
test.c  
C:\PROJECT\test\test.c(7) : error C2143: syntax error : missing ';' before 'return'  
Error executing cl.exe.  
  
test.exe - 1 error(s), 0 warning(s)
```

## #2

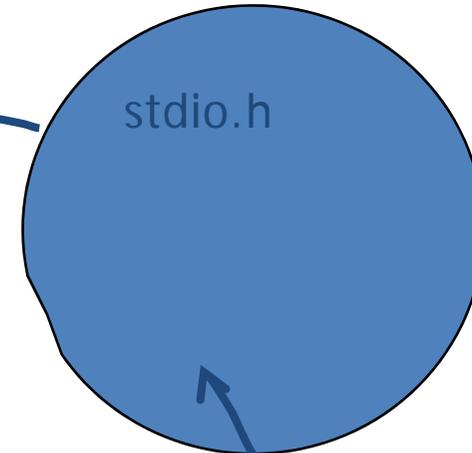
```
/*   가   *  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n")  
    return 0;  
}
```



```
-----Configuration: test - Win32 Debug-----  
Compiling...  
test.c  
c:\project\test\test.c(9) : fatal error C1071: unexpected end of file found in  
comment  
Error executing cl.exe.  
  
test.exe - 1 error(s), 0 warning(s)
```

# #3

```
/*          */  
#include <stdio,h>  
  
int main(void)  
{  
    print("Hello World!");  
    return 0;  
}
```

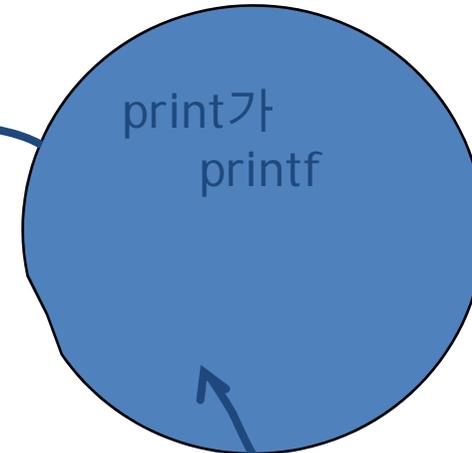


```
-----Configuration: test - Win32 Debug-----  
Compiling...  
test.c  
c:\project\test\test.c(2) : fatal error C1083: Cannot open include file: 'stdio,h':  
No such file or directory
```

# #4

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

int main(void)
{
    print("Hello World!");
    return 0;
}
```



```
-----Configuration: test - Win32 Debug-----
Compiling...
test.c
C:\CPROGRAM\test\test.c(6) : warning C4013: 'print' undefined; assuming extern
returning int
Linking...
test.obj : error LNK2001: unresolved external symbol _print
Debug/test.exe : fatal error LNK1120: 1 unresolved externals
Error executing link.exe.

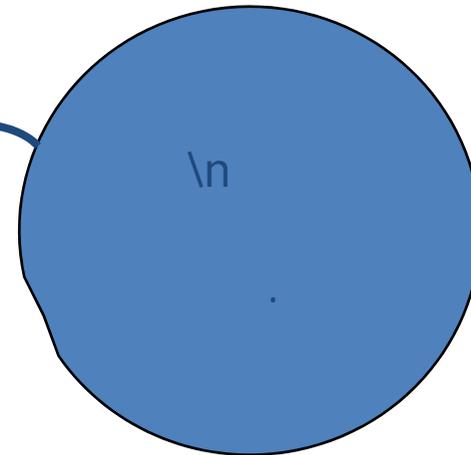
test.exe - 2 error(s), 1 warning(s)
```

가



# 가

```
/*          */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n"); //  
    printf("Good Morning\n");  
    return 0;  
}
```

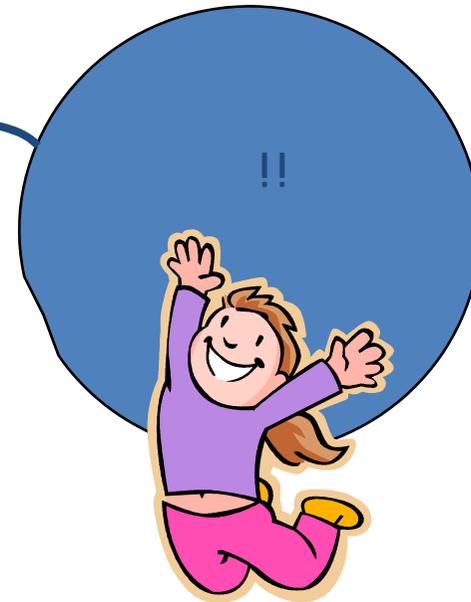


```
Hello World!  
Good Morning
```

# 가

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

int main(void)
{
    printf("Hello World! "); //
    printf("Good Morning\n");
    return 0;
}
```



Hello World! Good Morning

아무래도 이  
무문이 수상  
해..

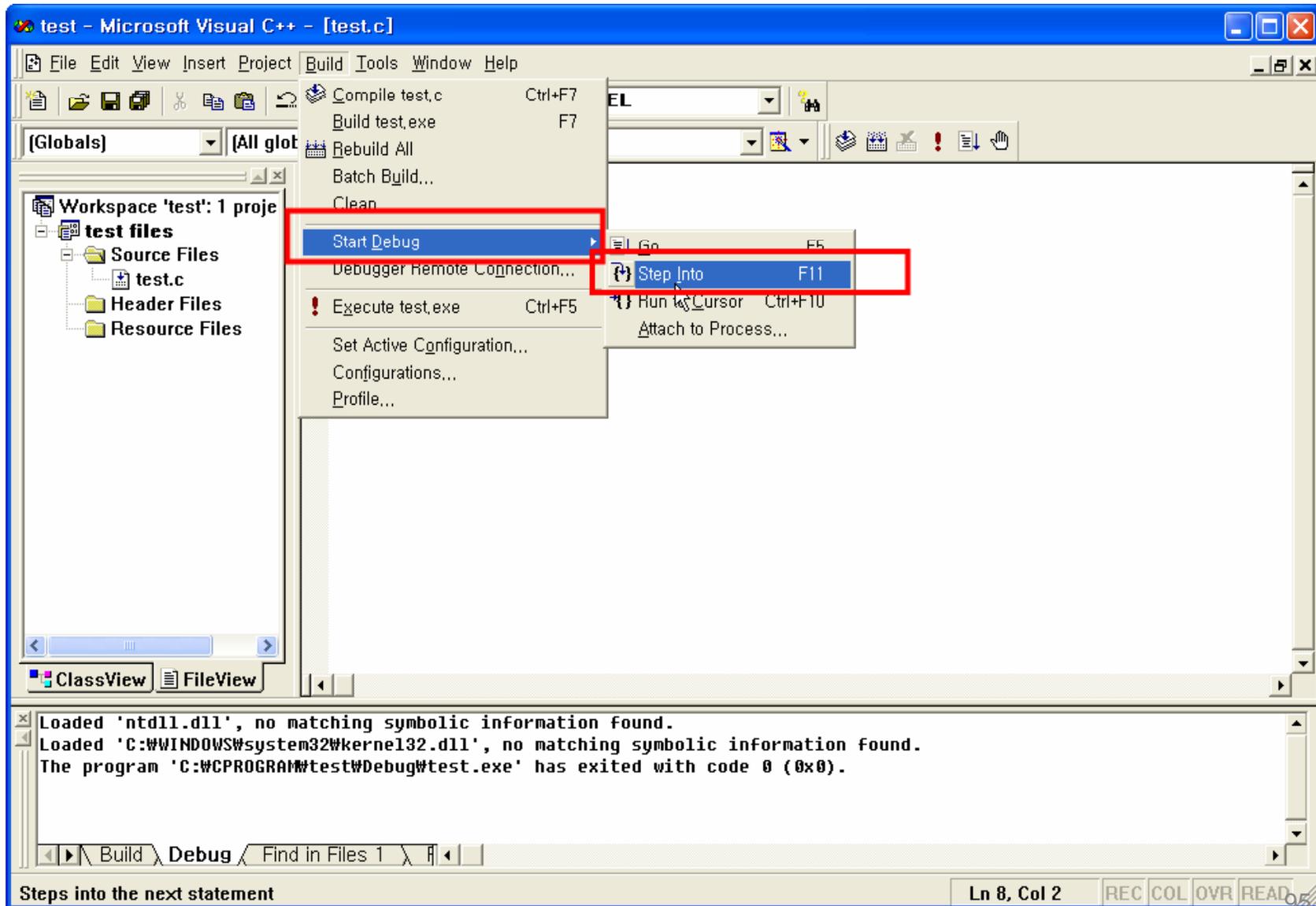


프로그램의  
실행결과

논리 에러를 발견  
하는 것은 수사관  
이 범죄 흔적을  
이용하여 범인을  
찾는 것과 같습니  
다.



# (debugger)



F10

test - Microsoft Visual C++ [break] - [test.c]

File Edit View Insert Project Debug Tools Window Help

GET\_PIXEL

[Globals] [All global members] main

```
/* 첫번째 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n");  
    return 0;  
}
```

Debug

C:\WCPROGRAMWtestWDebugWtest.exe

Context: main[]

Name	Value
Auto	Locals
this	

Loaded 'ntdll.dll', no matching sy  
Loaded 'C:\WINDOWS\system32\kerne

Build Debug Find in Files 1

Ready Ln 6, Col 1

F10

test - Microsoft Visual C++ [break] - [test.c]

File Edit View Insert Project Debug Tools Window Help

background

[Globals] [All global members] main

```
/* 첫번째 프로그램 */  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!");  
    return 0;  
}
```

Debug

C:\WPROJECTWtestWDebugWtest.exe  
Hello World?

Context: main()

Name	Value
@ printf returned	12

Auto Locals this Watch1 Watch2 Watch3 Watch4

Ready Ln 7, Col 1 REC COL OVR READ

- F5 (Go):
- F10 (Step Over): ( )
- F11 (Step Into): ( )
- F9 (Breakpoint):

# Q & A

