

TCT-시스템&솔루션개발 실기가이드

사번		성명	
유의 사항	1. 공정한 평가를 위해 동료들 도와주는 행위, 보여주는 행위를 금지하고 있습니다. 2. 부정행위 적발 시, 응시한 평가는 0점 처리됩니다. 3. 본 시험지는 응시장 외부로 유출할 수 없으며, 시험 종료 후 감독관에게 제출해야 합니다.		

- 1. 제공 파일 확인
 - 1) 제공 파일 다운로드 2
 - 2) 압축 풀기 2
- 2. 언어 별 개발 환경 세팅
 - 1) Java 언어 선택인 경우 3
 - 1-1) 개발 환경 세팅 3
 - 1-2) 개발 시 유의사항 5
 - 2) C 언어 선택인 경우 6
 - 2-1) cygwin -vi 사용 6
 - 2-2) eclipse C 사용 8
 - 2-3) 개발 시 유의사항 13
 - 3) C# 언어 선택인 경우 14
 - 3-1) 개발 환경 세팅 14
 - 3-2) 개발 시 유의사항 17
- 3. 제출 방법
 - 1) 작성한 답안 압축 18
 - 2) 제출 시스템(평가시스템) 접속 및 답안 제출 19
- 4. 설계용 연습장 20

1) 제공 파일 다운로드

- ① 바탕화면의 “제공파일” 폴더를 확인하여, 제공된 파일을 확인한다.
 - 응시하는 언어 별로 제공파일이 다르니, 파일명을 확인한 후 압축을 해제한다.
 - ✓ Java 선택 : **TCT_SP_JAVA_제공.exe** 파일을 더블클릭하여 압축을 해제한다.
 - ✓ C 선택 : **TCT_SP_C_제공.exe** 파일을 더블클릭하여 압축을 해제한다.
 - ✓ C# 선택 : **TCT_SP_CSHARP_제공.exe** 파일을 더블클릭하여 압축을 해제한다.
 - ✓ 실기 응시 중 “참고교재” 위치 내의 자료를 참고하는 것은 가능하다.

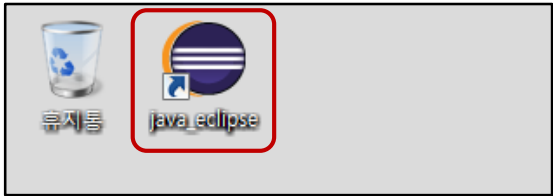
2) 압축 풀기

- ① 다운로드 받은 **TCT_SP_###_제공.exe** 파일을 더블 클릭하여 실행시키면, C:\Wsp_workspace 위치에 자동으로 제공파일의 압축이 해제된다.
- ② 제공된 내용을 확인한다.

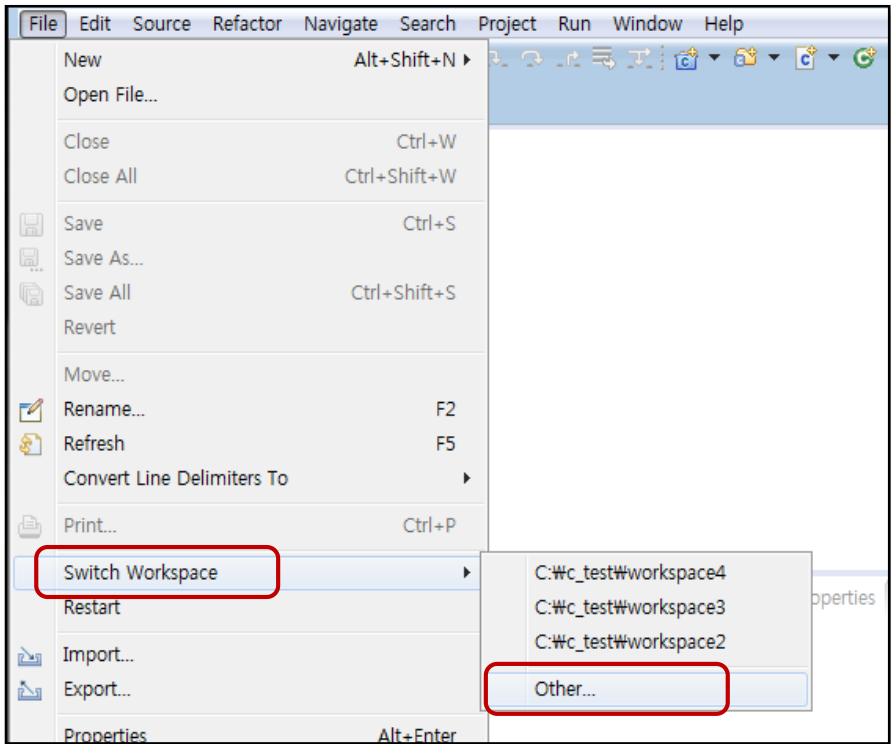
1) Java 언어 선택인 경우

1-1) 개발 환경 세팅

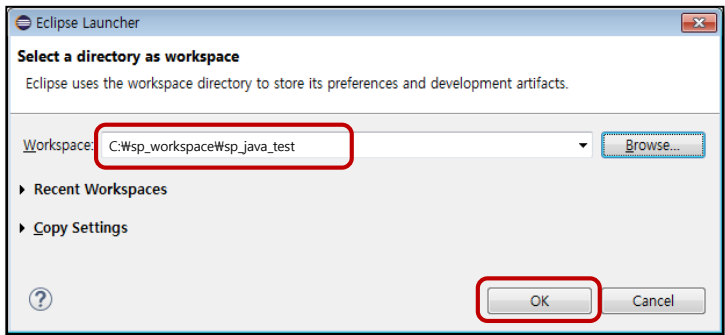
- ① 바탕화면에서 java_eclipse 아이콘을 더블 클릭하여 Java 개발 툴(tool)을 실행한다.
➔ 또는, C:\W\java_test\Weclipse 위치의 eclipse.exe 파일을 실행시켜도 된다.



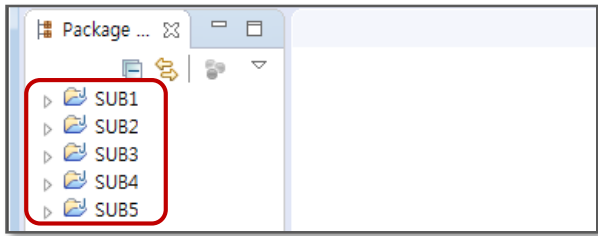
- ② eclipse에서 제공 파일로 다운로드 받은 workspace를 사용하기 위해서는 기존의 workspace를 변경해야 한다. File > Switch Workspace > Other 메뉴를 선택한다.



- ③ 다음과 같이 Workspace Launcher 창이 뜨면 Browse 버튼을 클릭하여 디렉토리 선택 창에서 “C:\Wsp_workspace\Wsp_java_test”를 선택하고, 확인 버튼을 클릭한 후 OK 버튼을 클릭한다.



④ eclipse가 자동으로 재시작되며, Package Explorer View 에 SUB1, SUB2, ... 프로젝트들이 보인다.

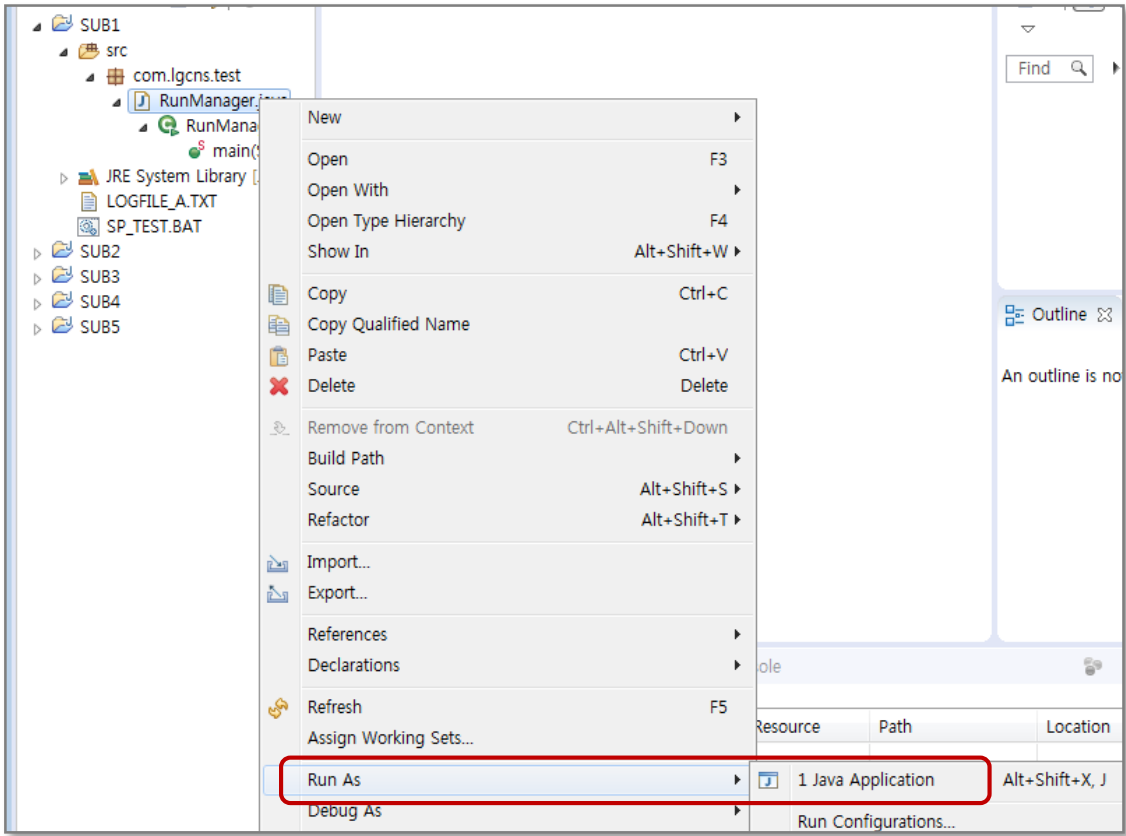


※ Package Explorer View 에 프로젝트가 보이지 않는 경우, Import 수행

⑤ 소문항 1번인 SUB1 프로젝트에서 프로그램 구현 후, Run 파일을 아래와 같이 실행하여 결과를 확인한다.

→ main() 가 있는 클래스를 선택한 후, 오른쪽 클릭 > Run As > Java Application 선택하여 실행

▼ 아래 그림에서 보이는 패키지명 및 클래스명은 문항마다 상이함



⑥ 실행 결과는 Console View 에 보이게 된다. (실행 결과는 문항마다 상이함)



1-2) 개발 시 유의사항

① 채점 시, 각 소문항별 프로젝트(SUB1, SUB2, ...)에서 SP_TEST.BAT를 실행함

* SP_TEST.BAT의 내용은 아래와 같음

(classpath가 프로젝트 하위의 'Wbin' 디렉토리와 기본 제공 Library 참조를 위해
'C:\Wsp_workspace\Wsp_java_test\Wlib' 디렉토리의 모든 jar 파일로 설정됨)

```
@echo off&setlocal
set base=%~dp0
set libdir=%~dp0..\lib
set classpath=%base%bin;%libdir%*
echo java -classpath %classpath% com.lgcns.test.RunManager
java -classpath %classpath% com.lgcns.test.RunManager
```

따라서, 제공된 com.lgcns.test.RunManager의 main() 에서 프로그램이 시작되도록 구현해야 함

별도의 외부 class들이나 jar 파일 사용시 SP_TEST.BAT의 java 커맨드 옵션에 해당 내용을 반드시 반영하여, 채점 시 정상적으로 동작할 수 있도록 해야 함

② SUB1부터 개발할 것을 권장하며, SUB2 이후의 내용은 이전 소문항 개발 내용(src)을 본인이 직접 복사하여 사용함. (eclipse 프로젝트 관련 파일들은 복사하지 말 것)

→ 문항 풀이 step : 소문항 1번(SUB1) 작성 → SUB1의 구현 내용 복사 → 소문항 2번(SUB2)에 SUB1 구현 내용 붙여넣기 → 소문항 2번 작성 → SUB2의 구현 내용 복사 → 소문항 3번(SUB3)에 SUB2까지의 구현 내용 붙여넣기 → ...

③ 제출 시, 컴파일 가능한 소스 파일이 각 소문항 별로 반드시 포함되어야 점수 인정함

④ SUB3, SUB4, SUB5 프로젝트에는 기본 참조 라이브러리 (gson, jetty)가 설정되어 있으며, 해당 라이브러리 파일은 "C:\Wsp_workspace\Wsp_java_test\Wlib" 디렉토리에 저장되어 있음.

※ Java API 참조

Java API 를 참고해야 하는 경우, 바탕화면의 Java API 아이콘을 더블 클릭한다.



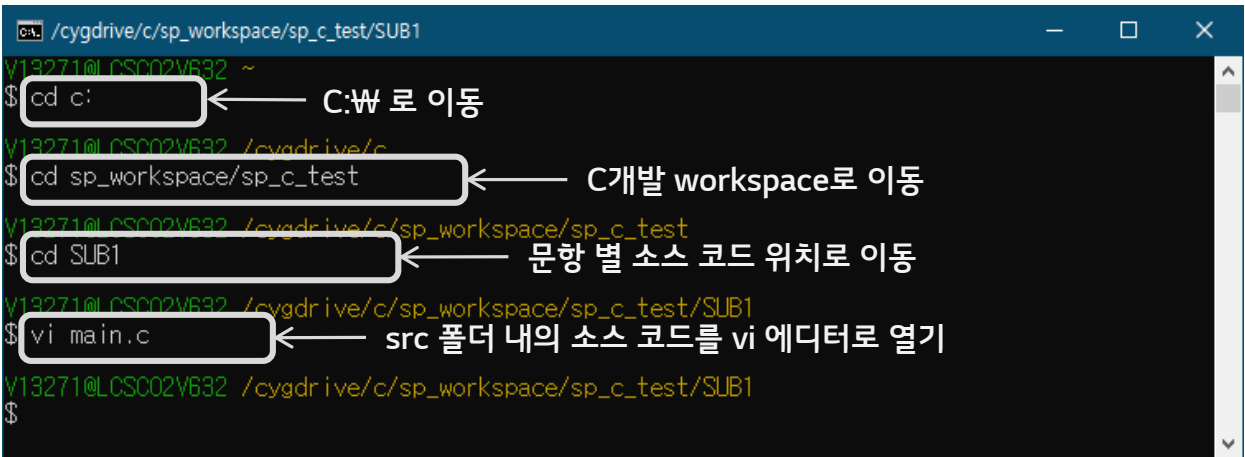
2) C 언어 선택인 경우

2-1) cygwin - vi 사용 (코드 편집 및 실행은 cygwin/eclipse 중 선택)

- ① 바탕화면에서 Cygwin Terminal 아이콘을 더블 클릭하여 bash terminal을 실행한다.
→ 또는, C:\Wc_test\Wcygwin 위치의 Cygwin Terminal 파일을 실행시켜도 된다.



- ② 아래와 같이 /cygdrive/c/sp_workspace/sp_c_test/SUB# 로 작업 디렉토리를 이동하여 vi 를 사용한다.

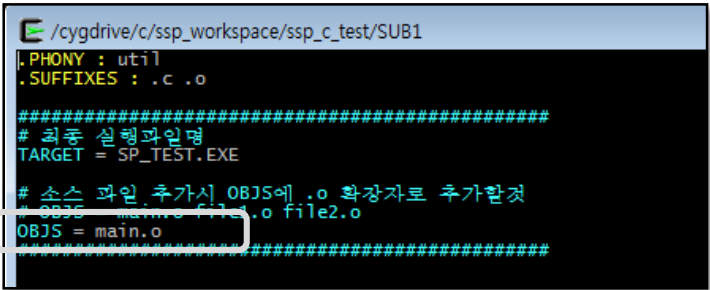


- ③ 소스 코드를 작성한 후, 컴파일은 각 소문항별 디렉토리(SUB1, SUB2, ...) 에 각각 존재하는 Makefile을 통해 이루어진다.
소스파일 추가/삭제, 외부 라이브러리 추가 등의 작업을 할 경우 기본 제공되는 Makefile을 수정한다.
Makefile내의 설명을 참고하여 원하는 내용을 추가/삭제한다.

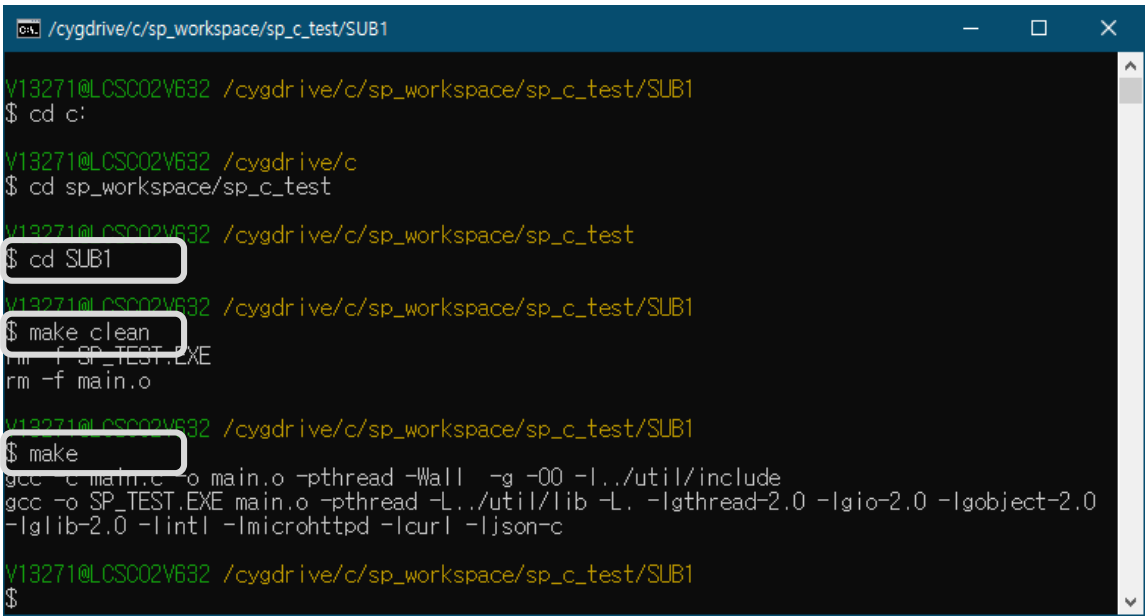
vi, eclipse, ultra edit 등으로 Makefile을 오픈한다.

```
$ cd c:/sp_workspace/sp_c_test/SUB1    ← 소문항 1번 SUB1 내의 Makefile을 수정할 경우
$ vi Makefile
```

- ④ Makefile내의 설명을 참고하여 원하는 내용을 수정한다.



- ⑤ Cygwin Terminal 을 이용하여 소스 코드를 컴파일하고 실행하는 방법은 아래와 같다.
- 각 소문항 별 디렉토리로 이동한다.
예) `$cd SUB1`
 - `make` 명령을 통해 `build`를 수행한다. 소스를 수정한 후, 필요할 때마다 `make`를 수행한다.
`$ make`
 - `clean` 빌드를 하고자 할 경우에는 아래와 같이 `make clean` 수행 후, 다시 `make` 를 수행한다.
`$ make clean`
`$ make`
 - `make` 가 성공적으로 종료되면 실행 프로그램(SP_TEST.EXE)이 생성된다.



```

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$ cd c:

V13271@LCS002V632 /cygdrive/c
$ cd sp_workspace/sp_c_test

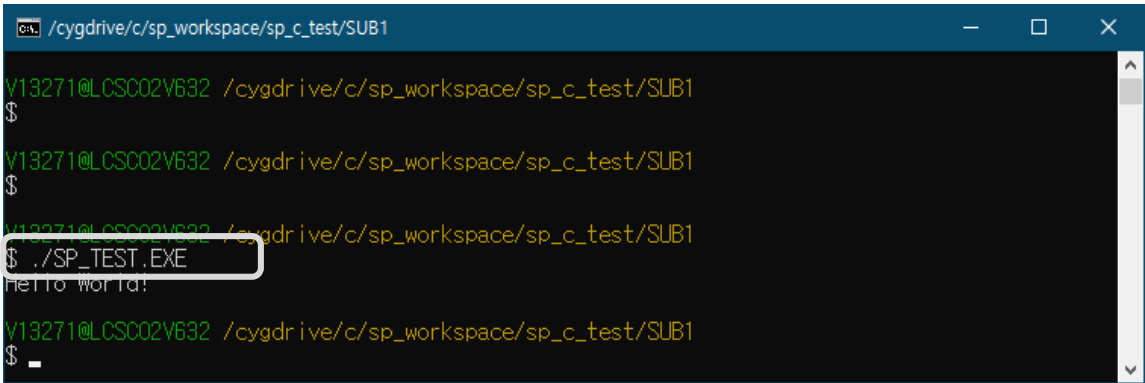
V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test
$ cd SUB1

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$ make clean
rm -f SP_TEST.EXE
rm -f main.o

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$ make
gcc -c main.c -o main.o -pthread -Wall -g -O0 -I../util/include
gcc -o SP_TEST.EXE main.o -pthread -L../util/lib -L. -lgthread-2.0 -lgio-2.0 -lgobject-2.0
-lglib-2.0 -lintl -lmicrohttpd -lcurl -ljson-c

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$
```

- Cygwin Terminal 에서 실행 시, 현재 디렉토리 표시(./)에 유의하여 아래와 같이 실행한다.



```

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$

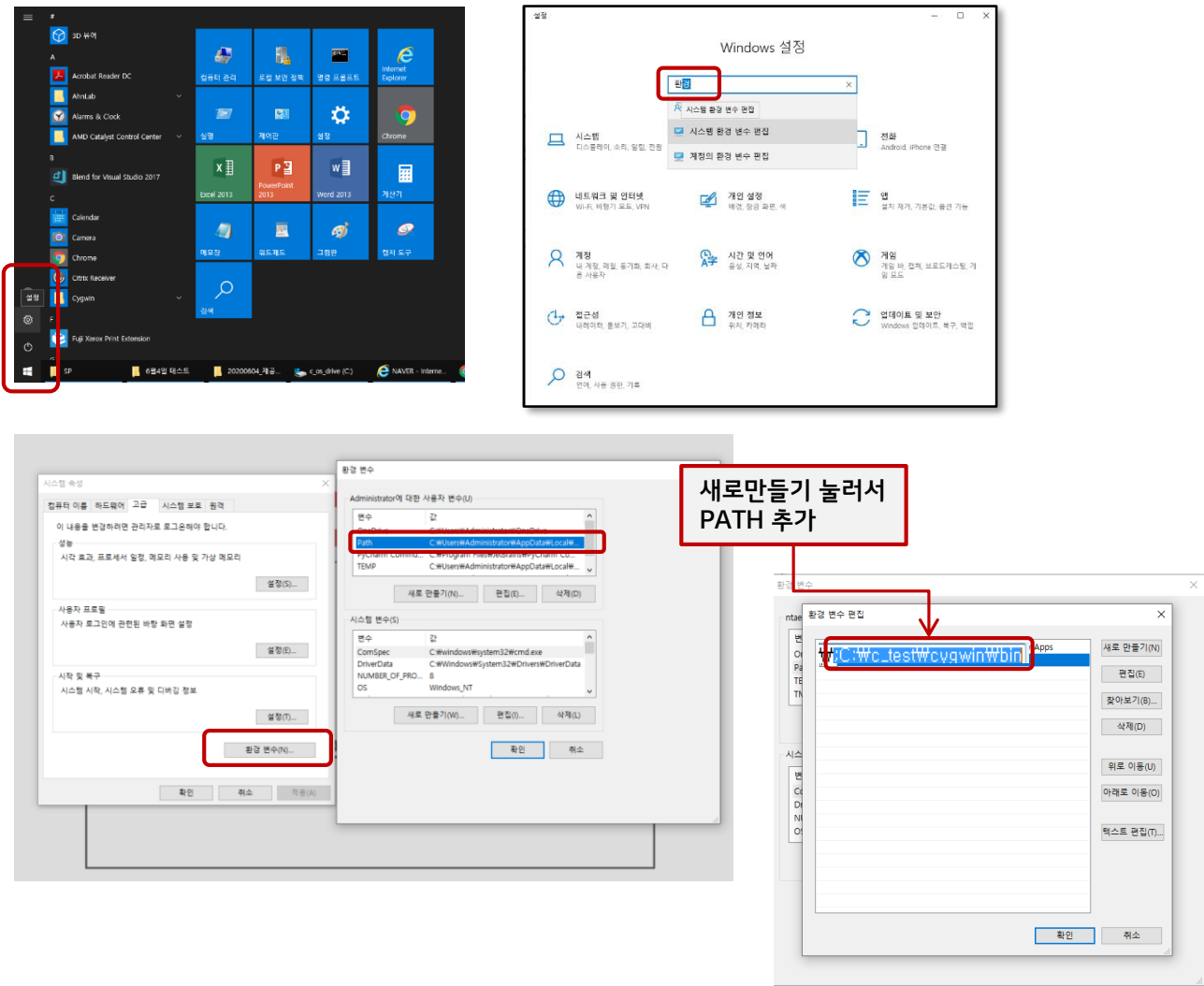
V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$

V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$ ./SP_TEST.EXE
Hello world!

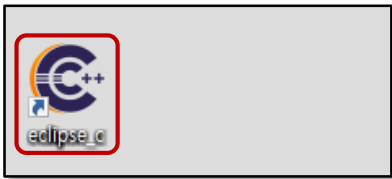
V13271@LCS002V632 /cygdrive/c/sp_workspace/sp_c_test/SUB1
$
```


2-2) eclipse C 사용 (코드 편집 및 실행은 cygwin/eclipse 중 선택)

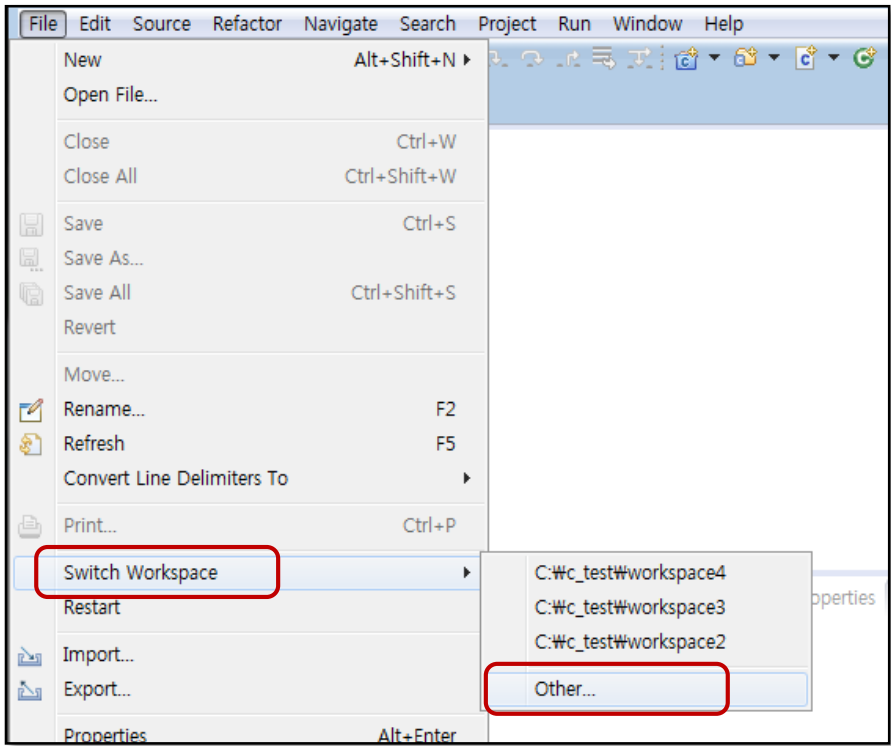
- ① eclipse 사용 전, 환경 변수를 잡아 주어야 한다.
(현재 환경에는 환경 변수가 잡혀 있으며, eclipse 내에서 컴파일이 잘 되지 않을 경우 해당 내용 수행)



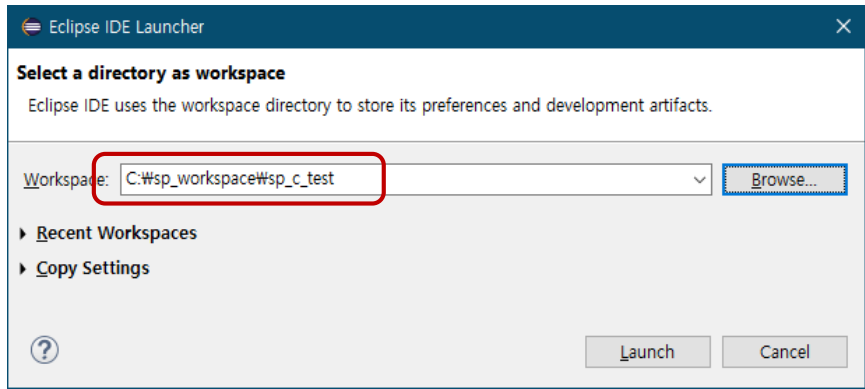
- ② 바탕화면에서 c_eclipse 아이콘을 더블 클릭한다. (반드시 환경 변수 잡아준 후 실행)
➔ 바탕화면에는 Java개발 테스트를 위한
java_eclipse도 있으므로, 알맞은 툴(tool)을 실행한다.



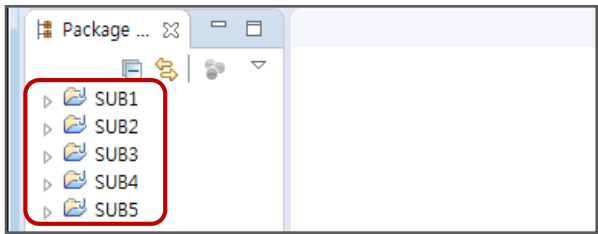
- ③ eclipse에서 제공 파일로 다운로드 받은 workspace를 사용하기 위해서는 기존의 workspace를 변경해야 한다. File > Switch Workspace > Other 메뉴를 선택한다.



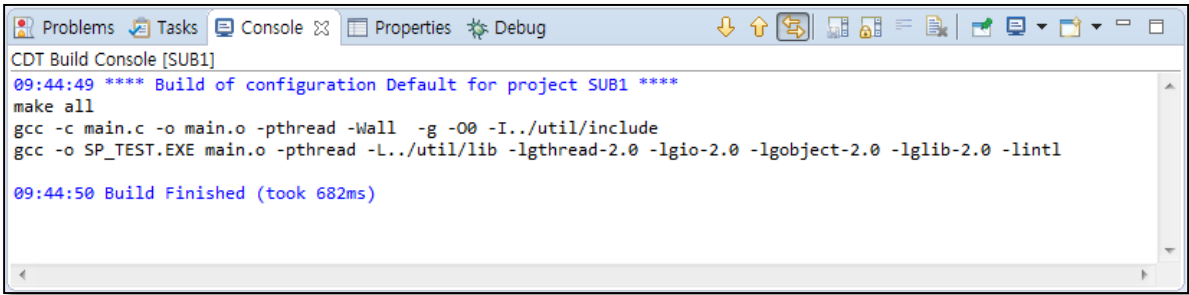
- ④ 다음과 같이 Workspace Launcher 창이 뜨면 Browse 버튼을 클릭하여 디렉토리 선택 창에서 “C:\wsp_workspace\wsp_c_test”를 선택하고, 확인 버튼을 클릭한 후 OK 버튼을 클릭한다.




- ⑤ eclipse가 자동으로 재시작되며, Package Explorer View 에 SUB1, SUB2, ... 프로젝트들이 보인다.

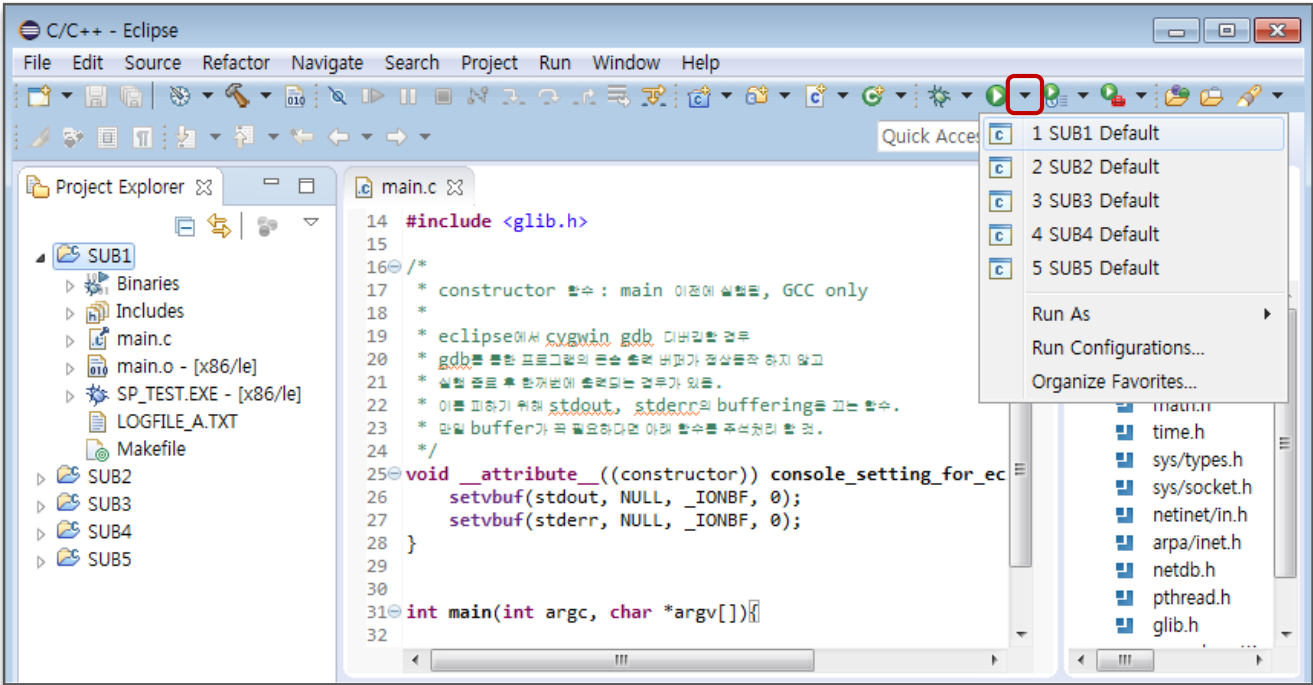


㉔ 컴파일 결과는 Console 창에 출력된다.



㉕ 프로그램 실행은 다음과 같이 할 수 있다.

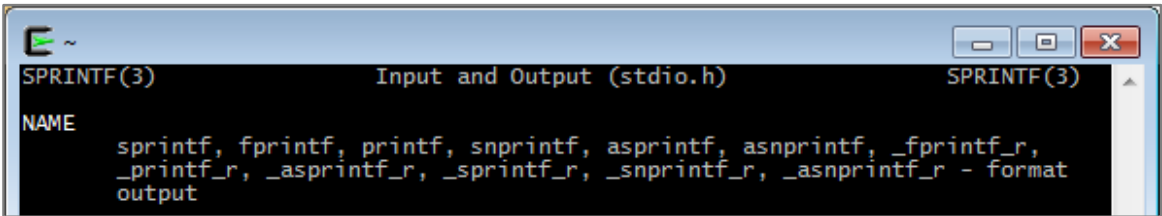
eclipse의  버튼의 아래 화살표를 클릭한 후 해당 소문항별로 SUB# Default 를 클릭하면 해당 프로그램이 실행된다. Argument 입력시에는 해당 Run Configuration을 수정하거나, cygwin 터미널에서 직접 실행할 수도 있다.



※ C API 참조 - man page

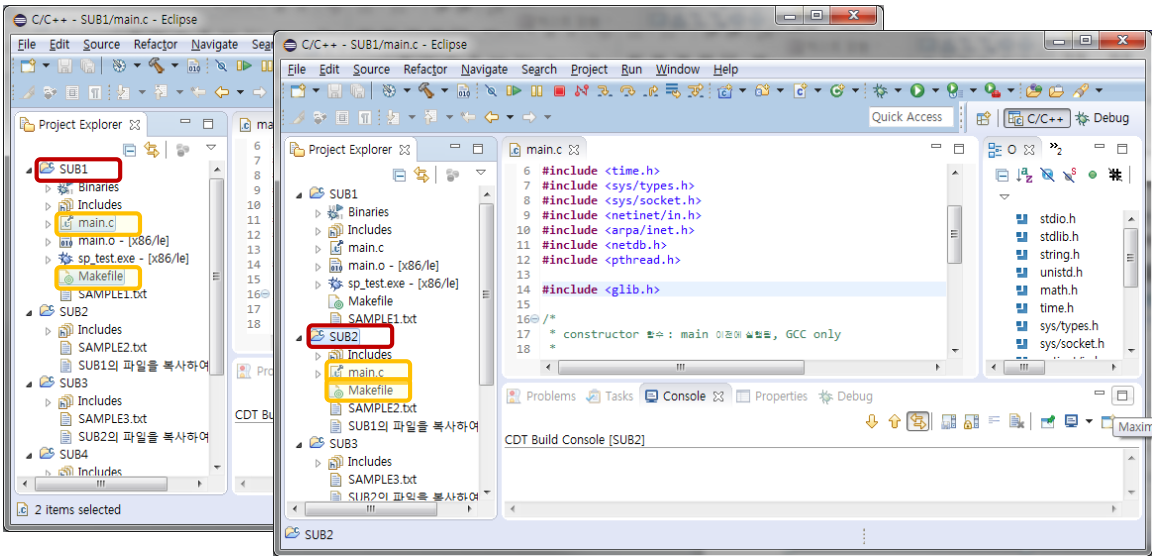
cygwin terminal 에서 man page 를 통해 C API 문서를 볼 수 있다.

\$ man 3 printf

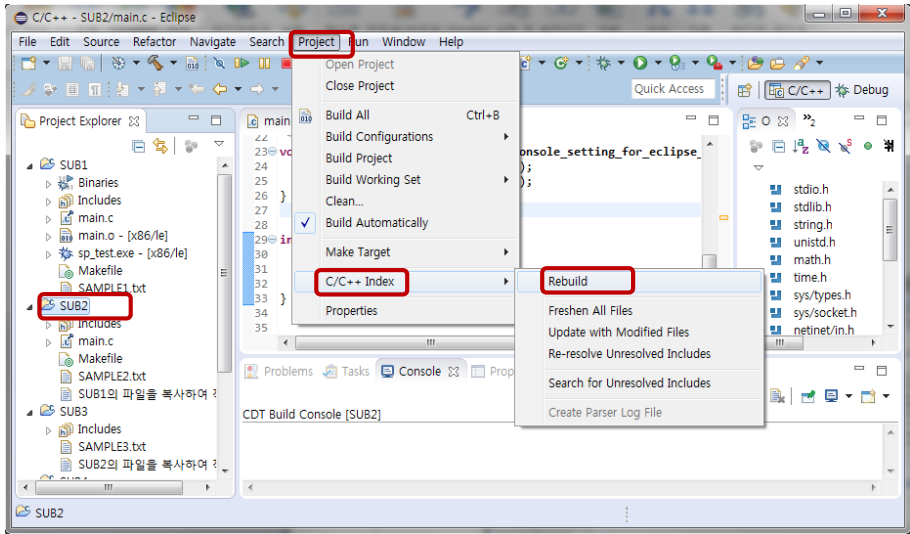


※ eclipse C 소문항 프로젝트간 파일 복사 가이드

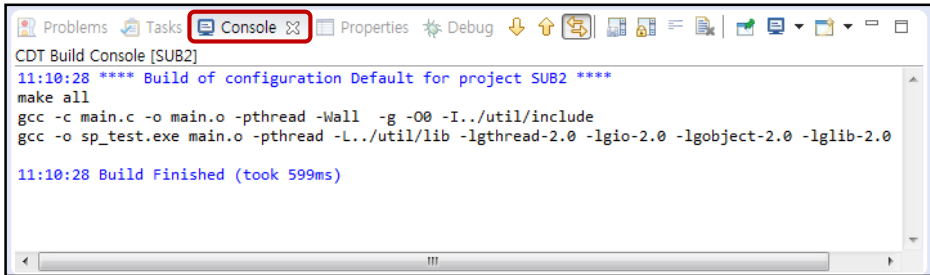
- ① 이전 소문항의 Makefile과 소스코드를 선택후 Ctrl + C 로 복사.
- ② 다음 소문항 프로젝트 선택후 Ctrl + V 로 붙여넣기.
(탐색기 등으로 직접 복사한 경우 F5를 누르면 복사된 파일이 나타남.)



- ③ 대상 프로젝트 선택된 상태에서 Project => C/C++ Index => Rebuild 수행
(Index를 Rebuild하면 복사된 파일의 소스코드상 빨간줄 대부분 제거됨.)



- ④ 프로젝트 빌드.



- * 일부 소스코드에 빨간줄이 보이더라도 빌드/디버그 정상 동작함. Console 탭에서 확인할 것.
- * Eclipse CDT / Cygwin GCC 환경에 익숙하지 않은 경우 Cygwin Terminal에서 빌드 권장.

2-3) 개발 시 유의사항

① 한글 인코딩 정보

- cygwin 기본 설정상 한글 출력은 UTF-8 인코딩 사용함
(프로그램에서 EUC-KR 한글코드를 화면(stdout) 출력시 깨질 수 있음)
- 기본 제공 파일의 모든 한글은 UTF-8 인코딩으로 되어있음.

② 채점 시 각 소문항별 디렉토리(SUB1, SUB2, ...)에서 SP_TEST.EXE를 실행할 것이므로,
폴더 구조, 실행파일의 위치 및 이름(SP_TEST.EXE)은 절대로 변경하여서는 안됨.
(기본 Makefile 사용시 해당 위치에 실행파일 생성됨)

③ SUB1부터 개발할 것을 권장하며, SUB2 이후의 내용은 이전 소문항 개발 내용을 본인이 직접 복사하여 사용함. (단, eclipse 프로젝트 관련 .project .cproject .settings 3개는 복사하지 말것)

➔ 문항 풀이 step : 소문항 1번(SUB1) 작성 → SUB1의 구현 내용 복사 →
소문항 2번(SUB2)에 SUB1 구현 내용 붙여넣기 → 소문항 2번 작성 →
SUB2의 구현 내용 복사 → 소문항 3번(SUB3)에 SUB2까지의 구현 내용 붙여넣기 → ...

④ 제출시 빌드 가능한 소스파일이 각 소문항별로 반드시 포함되어야 점수 인정함.

⑤ Eclipse 콘솔에 표준입력을 사용할 경우 개행문자에 Carriage-Return 문자가 포함됨을 유의할 것

⑥ 기본 제공 라이브러리 정보

- glib 2.50.3 (LGPL License)
C언어는 기본 포함되는 자료구조 기능이 없으므로, 언어간 형평성을 위해 glib 라이브러리를 기본 제공함 (single/double linked list, queue, hash table 등 포함)
- * 해더 파일 : util/include
- * 라이브러리 : util/lib
- * API 가이드 : util/doc/glib-2.0/index.html
- * 샘플 코드 : util/sample/glib_sample.c
- * 빌드 옵션 : 기본제공 Makefile에 설정되어 있음
- * 참고 URL : <http://developer.gnome.org/glib/2.50/>

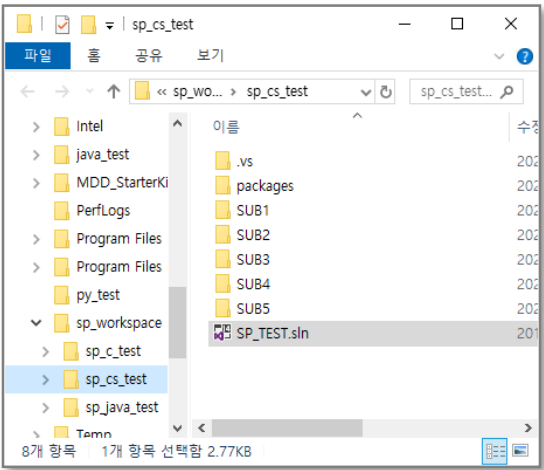
3) C# 언어 선택인 경우

3-1) 개발 환경 세팅

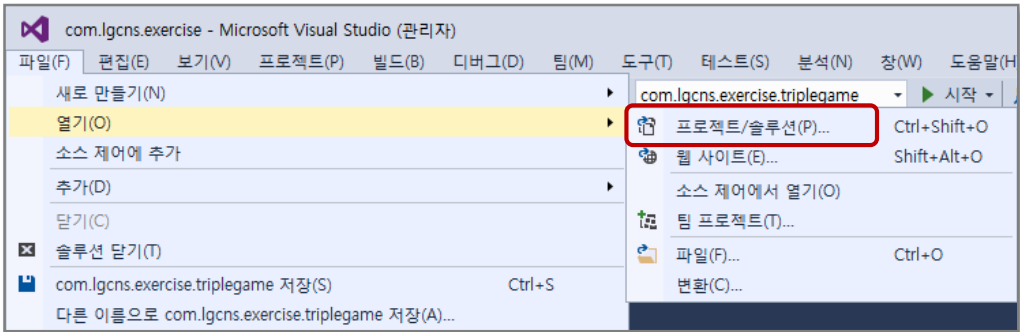
① 제공 파일의 솔루션(SP_TEST.sln) 파일을 실행한다.

최초로 Visual Studio 가 구동될 때에는
프로그램 로드 시간이 오래 걸린다.

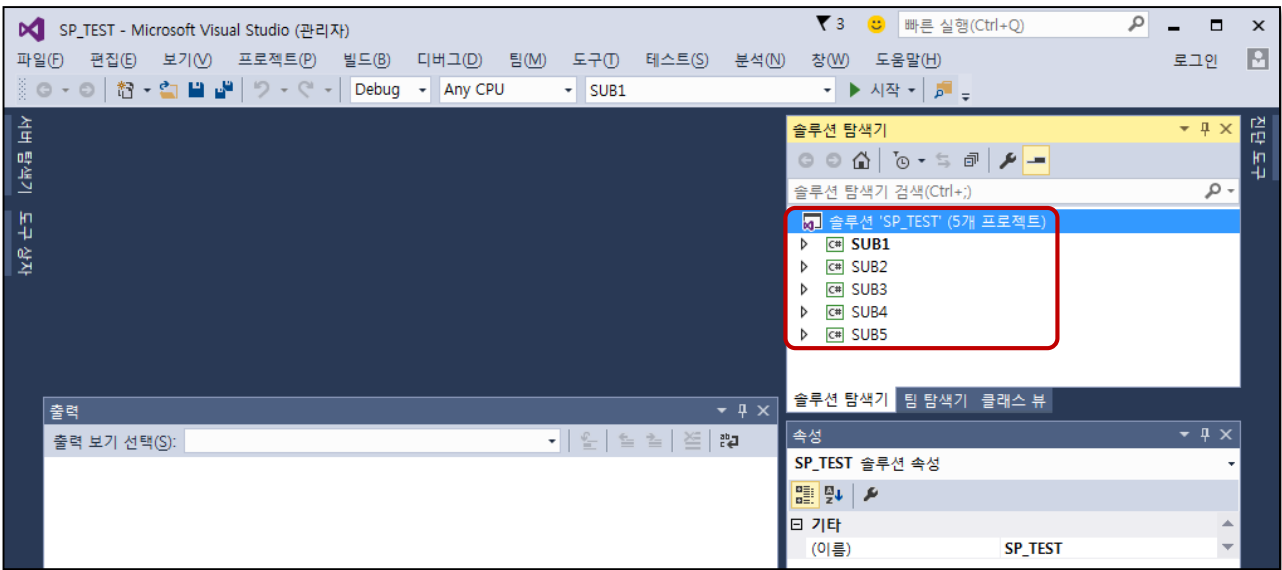
※ Visual Studio 로그인 창이 뜨는 경우,
“나중에 로그인” 링크를 클릭한다.



또는, 바탕화면의 Visual Studio 툴(tool)을 실행시킨 상태에서 파일 > 열기 > 프로젝트/솔루션 메뉴를
선택한다.



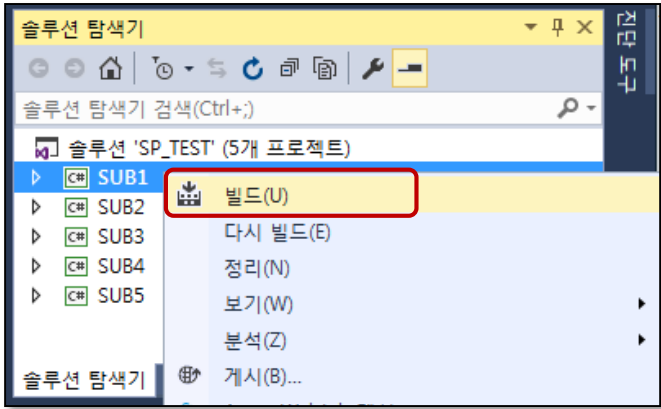
② 아래와 같이 솔루션 탐색기에 5개의 소문항 프로젝트(SUB1 ...)가 보여진다면 로드가 완료된 것이다.



③ C# 프로그램을 실행시키기 위해서는 "프로젝트 실행" 혹은 "파일 직접 실행"을 통해 구동한다.

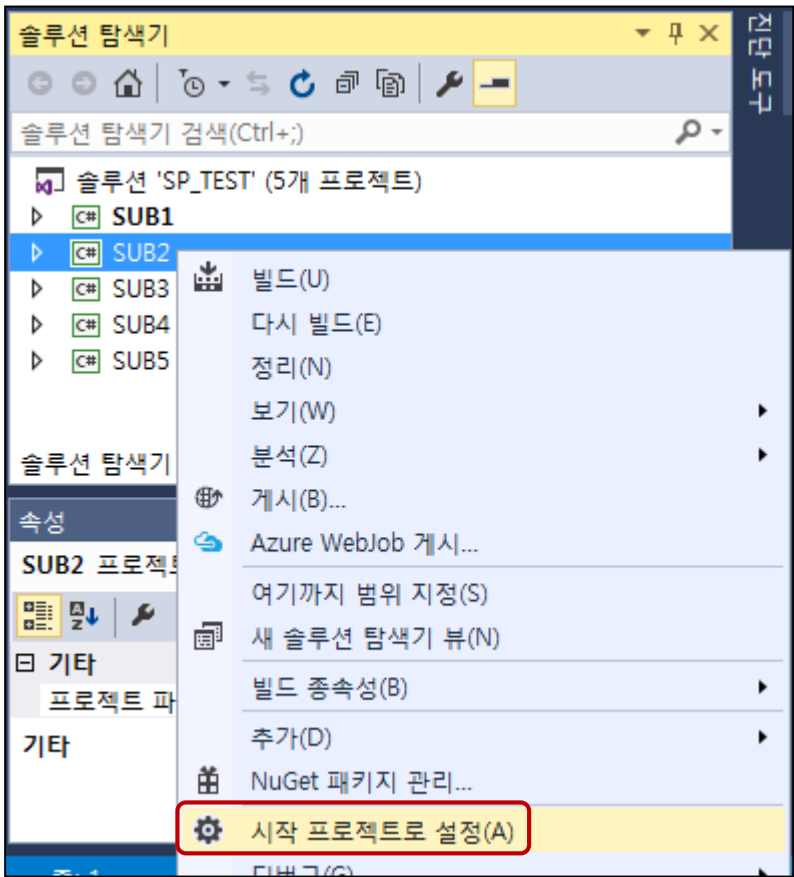
[방법 1. 빌드 후 실행]

빌드 후 각 프로젝트 하위 Bin 폴더에 빌드 모드별로 생성된 파일 실행
(프로젝트명에 우측 버튼 선택 시 파일 탐색기에서 폴더 열기 제공)

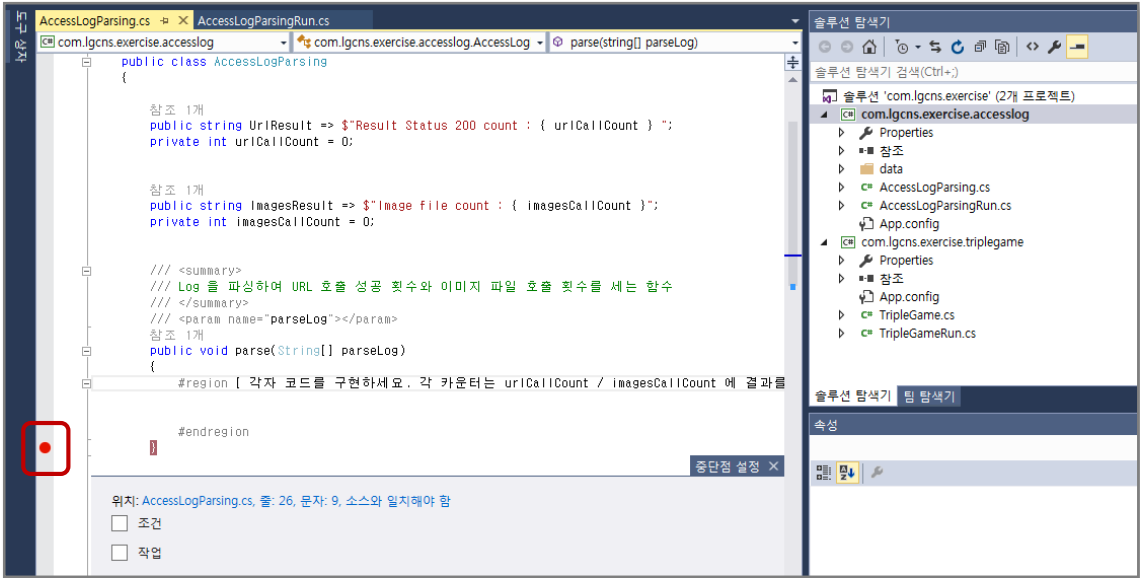


[방법 2. 시작 프로젝트로 설정하여 실행]

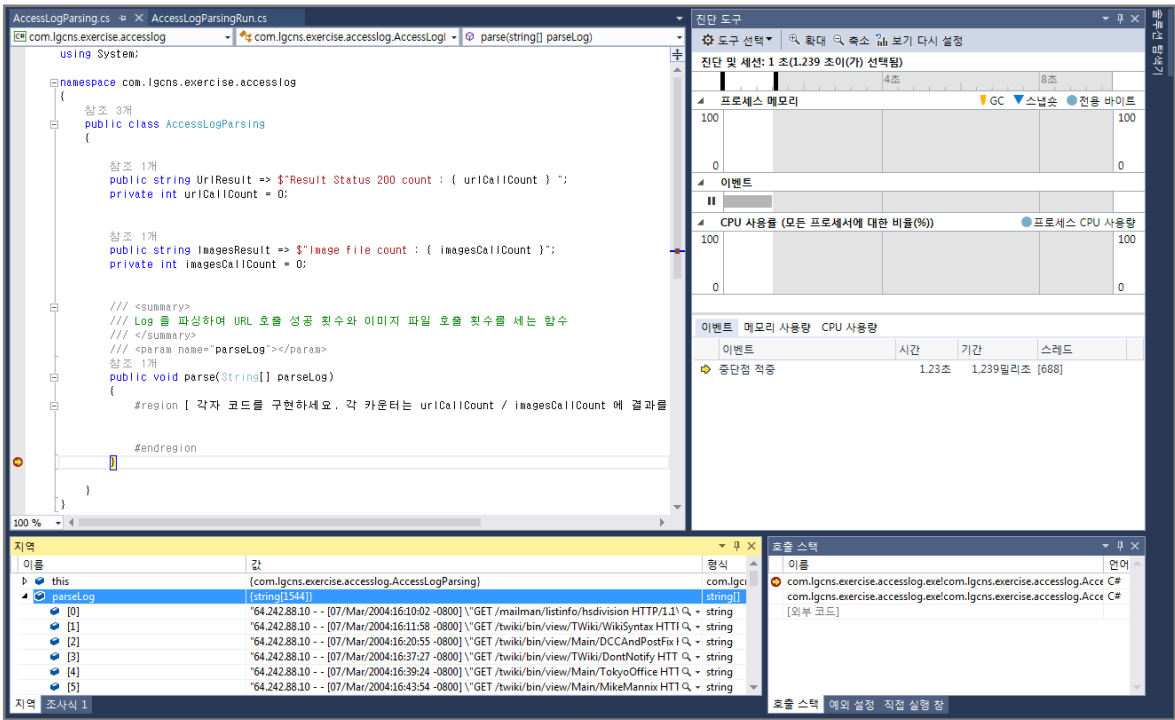
실행 버튼을 눌러서 시작프로젝트로 설정된 프로젝트를 바로 실행 (5개 프로젝트 중 시작 프로젝트로 설정된 프로젝트명은 bold로 표시됨)



- ④ 프로그램을 디버깅하기 위해서는 디버그 실행(F5) 을 선택한다.
에디터 창의 왼쪽 회색 바를 선택하게 되면 Node 별 중단점을 설정 가능하다.



- ⑤ 디버그 모드로 실행 시 Exception 이 발생하거나 중단점이 설정된 부분에서는 아래와 같이 화면이 전환되며, 디버깅을 할 수 있도록 제공된다.



3-2) 개발 시 유의사항

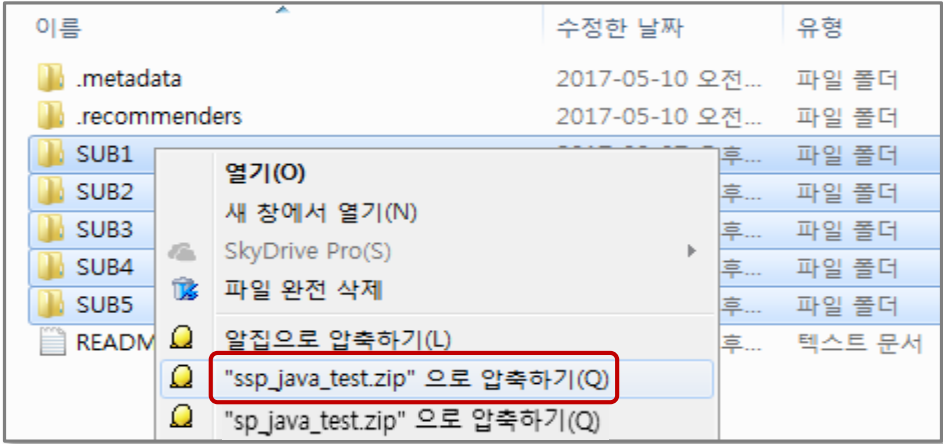
- ① 채점 시 각 소문항별 폴더(SUB1, SUB2, ...)에서 SP_TEST.exe를 실행할 것이므로, 폴더 구조, 실행파일의 위치 및 이름(SP_TEST.exe)은 절대로 변경하여서는 안됨.
(제공된 Visual Studio프로젝트에 '각 소문항별 폴더 바로 아래에 SP_TEST.exe가 생성되도록' 설정되어있음.)
- ② SUB1부터 개발할 것을 권장하며, SUB2 이후의 내용은 이전 소문항 개발 내용을 본인이 직접 복사하여 사용함.

→ 문항 풀이 step : 소문항 1번(SUB1) 작성 → SUB1의 구현 내용 복사 → 소문항 2번(SUB2)에 SUB1 구현 내용 붙여넣기 → 소문항 2번 작성 → SUB2의 구현 내용 복사 → 소문항 3번(SUB3)에 SUB2까지의 구현 내용 붙여넣기 → ...

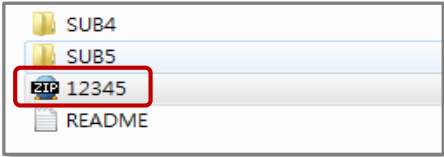
- ③ 제출 시 빌드 가능한 소스 파일이 각 소문항 별로 반드시 포함되어야 점수 인정함.

1) 작성한 답안 압축

- ① 파일 제출을 위해 작성한 SUB1 ~ SUB5 폴더를 한꺼번에 압축한다.
→ 아래의 예는 Java언어를 선택하여 C:\Wsp_workspace\Wsp_java_test 폴더에서 선택한 경우임

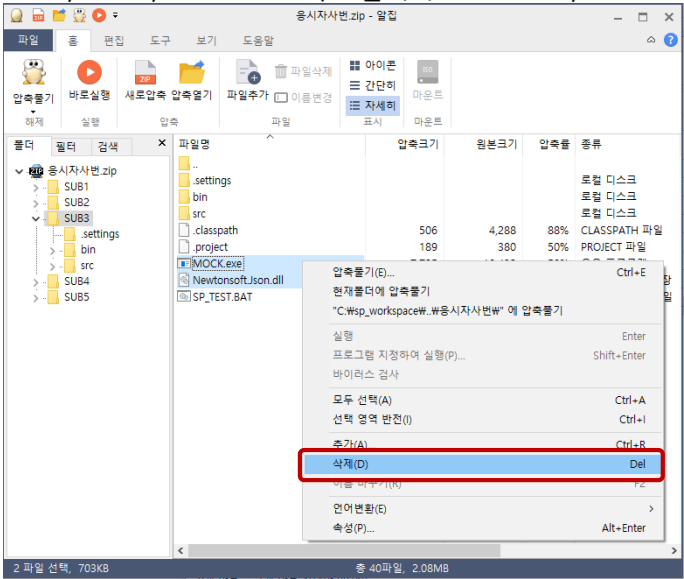


- ② 압축 파일명을 사번.zip 으로 수정한다.



- ③ 압축 파일 사이즈 확인

평가시스템에 업로드 시, **최대 10MB**까지만 업로드 가능하므로 압축 파일의 용량을 체크한다.
10MB 가 초과하는 경우 사번.zip 을 **알집으로 오픈(더블클릭)**한 후 아래 파일들을 삭제한다.
※ SUB3, SUB4, SUB 5 프로젝트 폴더의 MOCK.exe, Newtonsoft.Json.dll 삭제



※ 별도의 외부 Library를 사용하여 10MB를 초과한 경우 동일 방식으로 해당 Library를 삭제하고, 시험감독관에게 '외부 라이브러리'를 사용하였음을 알려준다.

2) 제출 시스템(평가시스템) 접속 및 답안 제출

