

문제해결을 위한

창의적 알고리즘(중급)



■ 집필진

안성진(성균관대학교 교수)
송태옥(가톨릭관동대학교 교수)
장승연(성균관대학교 연구원)
정종광(경기과학고등학교 교사)
배준호(경남정보고등학교 교사)
김봉석(경남과학고등학교 교사)
오은희(창원과학고등학교 교사)
정혜진(경기과학고등학교 교사)
전현석(경기과학고등학교 교사)
문광식(세종특별자치교육청 교사)
장원영(충북교육정보원 교사)
최웅선(수원정보과학고등학교 교사)
이 진(인천과학고등학교 교사)
박병기(서울과학고등학교 교사)
김종혜(경기과학고등학교 교사)
한건우(경기모바일과학고등학교 교사)
정웅열(경기북과학고등학교 교사)
배성환(광주과학고등학교 교사)
박소영(부산일과학고등학교 교사)

■ 목 차 ■

I. 문제해결과 알고리즘을 위한 준비 단계

1. 무료 C언어 통합개발환경 기본 사용 방법	7
가. Ubuntu Linux && Code::Blocks	8
나. Windows && Orwell DevC++	12
다. Mac OS X && Xcode	15
라. 파일입출력 기본	18
2. 정보과학과 문제	20
가. 계산 문제	20
나. 결정 문제	21
다. 최적화 문제	21
3. 알고리즘과 실행시간 측정	22
가. 알고리즘	22
나. 실행시간의 측정	23

II. 탐색기반 알고리즘의 설계

4. 탐색	37
가. 선형구조의 탐색	39
나. 비선형구조의 탐색	67
5. 전체탐색법	107
가. 선형구조와 비선형구조의 전체탐색	107
6. 탐색공간의 배제	254
가. 수학적 배제를 이용한 설계	254
나. 경험적 배제를 이용한 설계	280

Part

I

문제해결과 알고리즘을 위한 준비 단계

1. 무료 C언어 통합개발환경 기본 사용 방법
2. 정보과학과 문제
3. 알고리즘과 실행시간측정

문제해결과 알고리즘을 위한 준비 단계



이 교재는 C/C++언어를 1년 이상 다룬 학생으로 기본적인 프로그래밍 언어의 문법은 모두 아는 학생을 대상으로 한다. 따라서 기본적인 프로그래밍 언어의 문법에 대한 설명은 하지 않으며, 알고리즘 설계를 위주로 구성된 교재이다.

이 단원에서는 무료로 이용할 수 있는 C/C++언어 통합개발환경의 설치 및 기본적인 사용법에 대해서 먼저 소개하고, 정보과학에서 다루는 문제의 특성 및 이를 해결하기 위하여 설계된 알고리즘의 성능을 분석하는 방법에 대해서 다룬다.

1 무료 C언어 통합개발환경 기본 사용 방법

프로그래밍 언어를 활용하는 문제 해결을 위해서는 자신이 생각한 문제 해결 방법과 알고리즘을 프로그래밍 언어로 작성하고 실행한 후, 출력되는 결과를 자신이 예상한 결과와 비교해 보는 과정이 필요하다.

가장 많이 사용되는 프로그래밍 언어 중 하나인 C/C++ 코드를 작성하고, 실행 결과를 확인하고, 실행파일로 만들 수 있는 많은 통합개발환경(IDE)이 있지만, 상업적 판매 목적의 소프트웨어를 만들어내기 위한 큰 프로젝트가 아닌 이상 고가의 유료 통합개발환경을 구입해서 사용할 필요는 없다.

C/C++ 언어로 작성된 코드들은 일반적으로, 국제적으로 제정되어있는 표준 방법들을 사용하면 모든 통합개발환경에서 정상적으로 컴파일 되고 동작하기 때문이다.

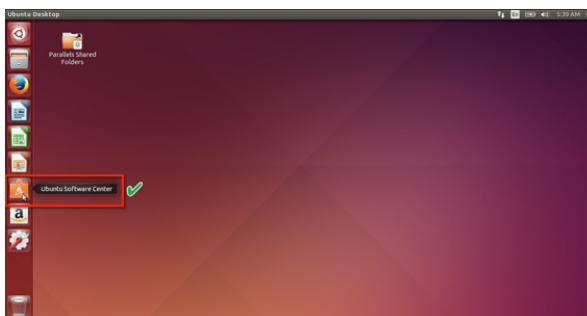
다양한 프로그램들이 있지만, 무료로 사용할 수 있으면서도 편리하고 강력한 몇 가지 C/C++ 통합개발환경에 대해서 살펴보자.

가. Ubuntu Linux && Code::Blocks

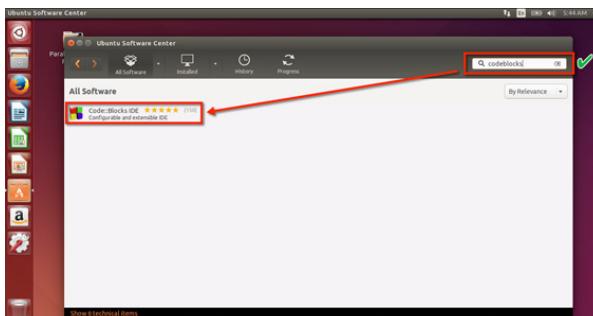
코드블락스(Code::Blocks)는 매우 강력하면서도 지속적으로 버전업되고 있는 무료 공개 IDE로, Windows/Linux/MacOSX 운영체제에서 모두 사용할 수 있다.

Ubuntu Linux에 Code::Blocks를 설치하고 간단히 실행하는 방법은 다음과 같다.

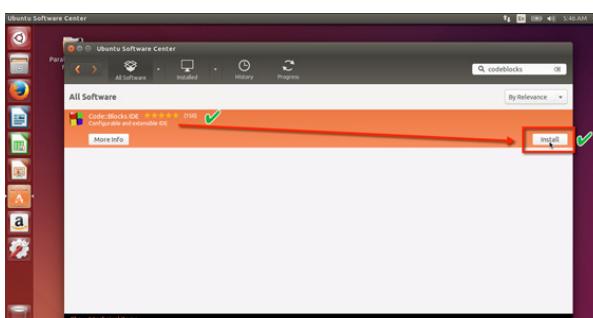
(공식사이트 : <http://www.codeblocks.org/>에서는 운영체제에 따른 설치 파일을 제공하고 있다.)



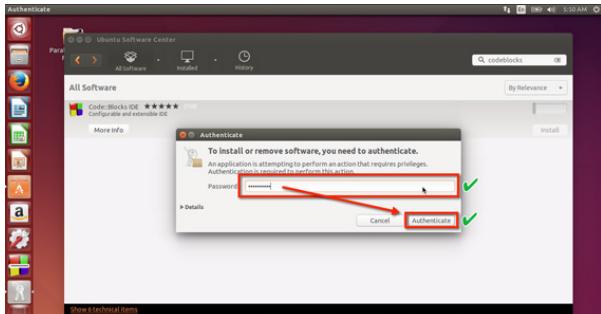
우분투 운영체제에서 왼쪽 메뉴의 우분투 소프트웨어 센터(Ubuntu Software Center)를 클릭한다.



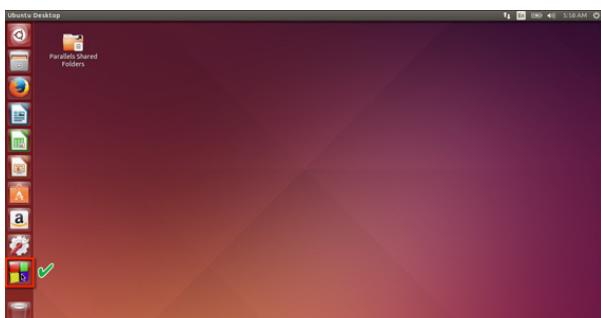
검색창에서 **codeblocks**를 검색한다.



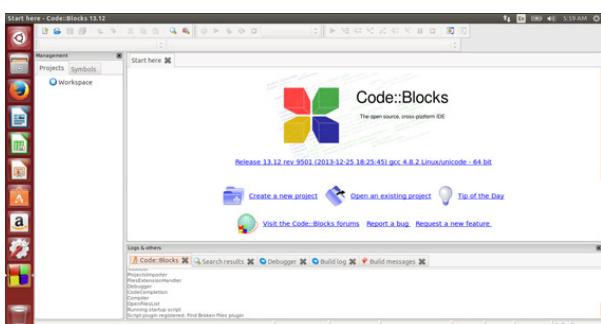
codeblocks를 누른 후 활성화되는 **install**을 누른다.



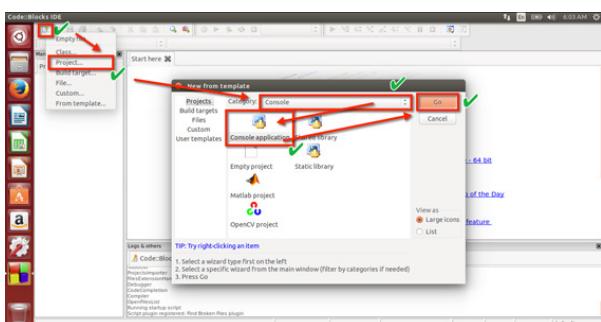
설치를 위해 비밀번호를 입력하고 설치를 진행한다.



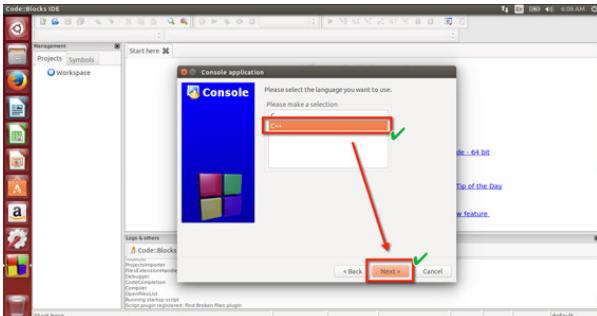
설치가 완료되면 왼쪽 메뉴에서 코드블락스 아이콘을 눌러 실행시킨다.



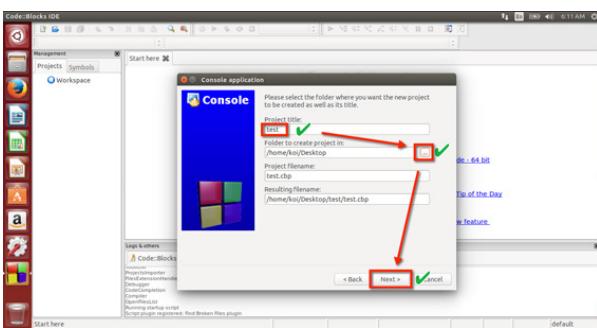
코드블락스 실행을 확인한다.



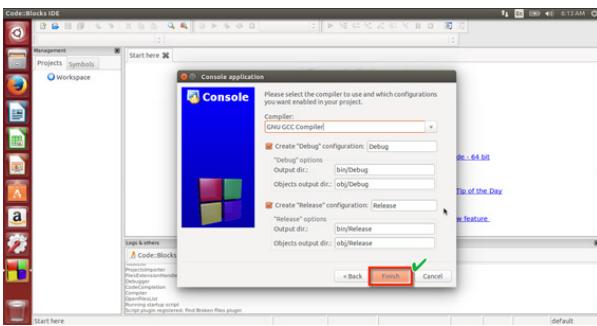
프로젝트(폴더)를 만들기 위해 **New Project**를 눌러 실행하고, **Category**에서 **Console**을 선택한다. 아래에서 **Console application**을 선택하고 **GO**를 누른다.



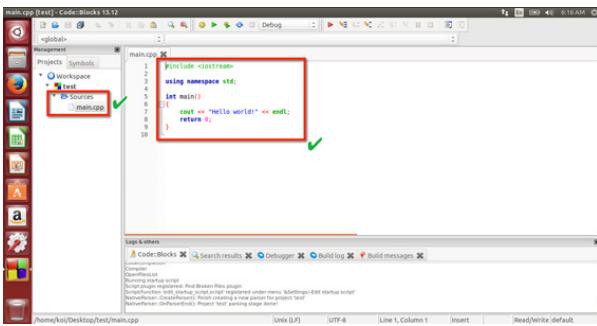
작성할 언어 종류를 C++로 선택하고 **Next**를 누른다.



프로젝트 폴더명, 저장할 위치/Desktop은 처음 바탕화면을 의미)를 선택하고 **Next**를 누른다.



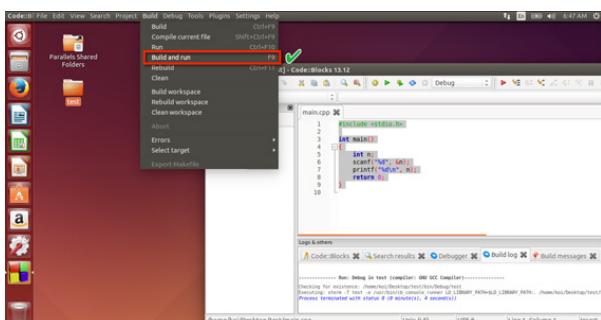
Finish를 누르면 표준 GCC 컴파일러로 설정이 되고, 코드 작성 준비가 끝난다.



왼쪽 탐색 창에서 Sources 폴더에서 main.cpp 파일을 놀려보면 기본 C++ 코드가 들어있고 편집할 수 있게 된다.

아래와 같은 기본 코드를 작성해 넣어보자.

줄	코드	참고
1	#include <stdio.h>	
2		
3	int n;	
4		
5	int main()	
6	{	
7	scanf("%d", &n);	
8	printf("%d\n", n);	
9	return 0;	
10	}	

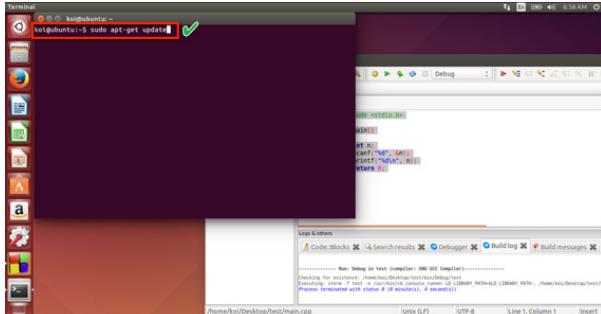


작성한 코드를 컴파일하고 실행하기 위해서는 코드블럭스 메뉴에서 Build – Build and run을 누르면 된다.

처음 설치하였을 때에는 g++ not found 와 같은 컴파일 오류가 발생할 수 있다. 그럴 때에는 터미널 명령 입력 창을 열고 몇 가지 명령을 실행시키면 해결된다.

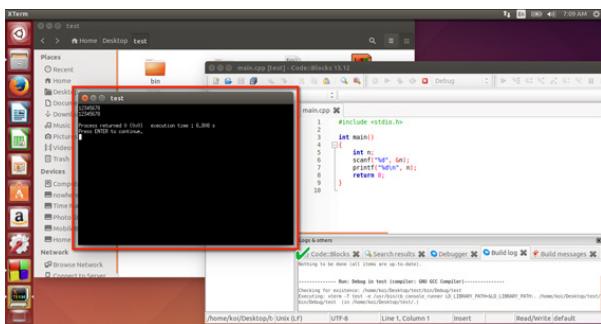


Dash home을 실행하고 검색창에서 terminal을 찾아 실행시킨 후



명령 입력 창에서
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential
gcc -v
make -v

를 순서대로 실행하면 코드블락스에서 작성한 소스코드를 컴파일/실행 가능



Build - Build and run을 누르면

실행창으로 실행이 되면서 입출력이 가능하게 된다.

4. Windows & Orwell DevC++

Orwell DevC++는 무료 공개 IDE인데 Windows 운영체제에서만 사용할 수 있다. 한국 KLDP 그룹에 의해 한글화가 제공되어, 한글 메뉴를 사용할 수 있고, 기본 설치만으로도 한글이 포함된 소스코드를 사용할 수 있다.

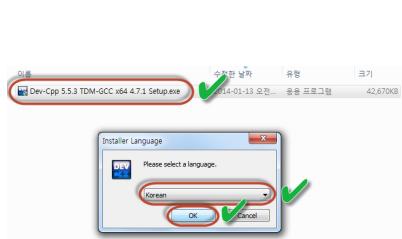
Windows에 Orwell DevC++를 설치하고 간단히 실행하는 방법은 다음과 같다(공식 사이트 : <http://orwelldevcpp.blogspot.kr/>에서는 몇 가지 다른 설치 파일을 제공하고 있는데, 일반 버전은 <http://sourceforge.net/projects/orwelldevcpp/>를 통해 다운로드하고 설치 할 수 있다.).



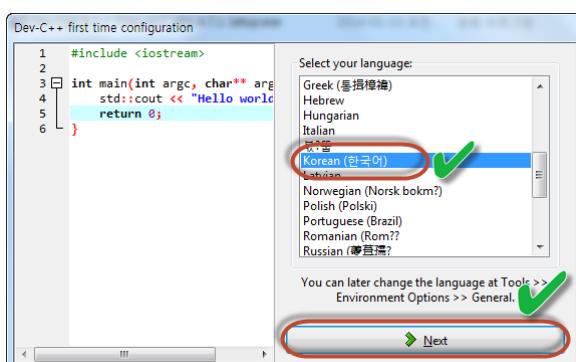
orwell dev c++를 검색하고 다운로드 사이트를 누른다.



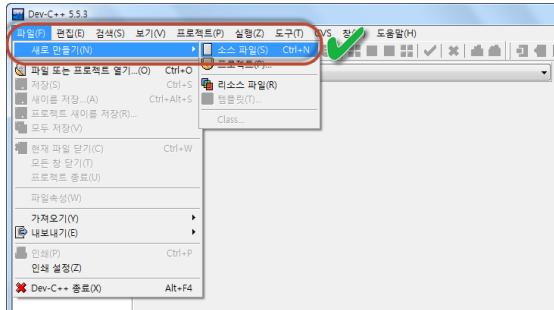
공식 배포 사이트를 통해 설치 파일 다운로드 링크를 누른다.



설치 언어로 한국어 설정

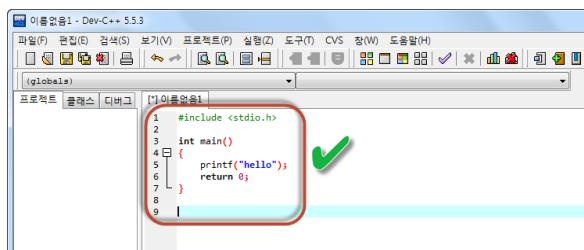


기본 언어로 한국어를 선택하여 설치 완료



프로그램을 실행시킨 후

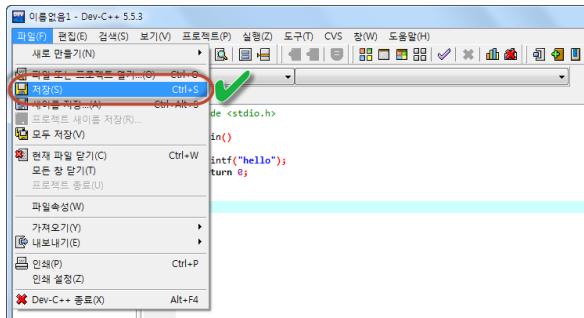
파일 – 새로 만들기 – 소스 파일을 눌러 코드를 작성할 수 있다.
(프로젝트를 만들지 않아도 소스 파일만 작성하고도 실행이 가능하다.)



소스 코드를 작성한다.

아래와 같은 기본 코드를 작성해 넣어보자.

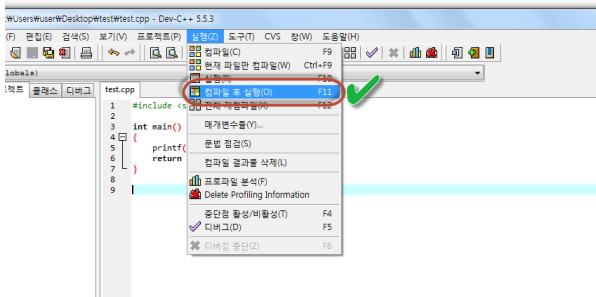
줄	코드	참고
1	#include <stdio.h>	
2		
3	int main()	
4	{	
5	printf("hello");	
6	return 0;	
7	}	



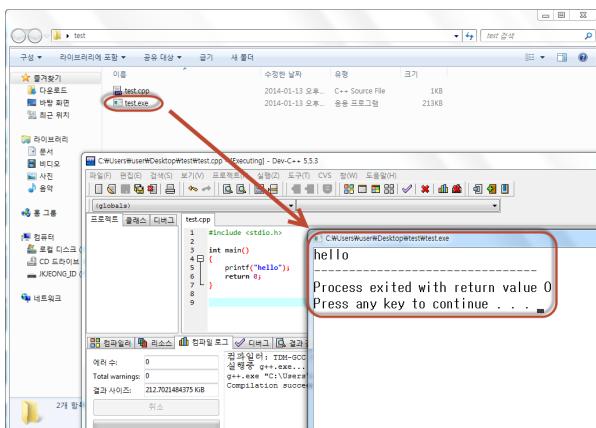
소스 코드를 파일로 저장한다.

실행 - 컴파일 후 실행

을 누르면 실행 파일이 만들어지고 실행 결과를 확인할 수 있다.



실행 결과 확인



4. Mac OS X && Xcode

Xcode는 무료 IDE인데 Mac OS X 운영체제에서만 사용할 수 있다. Xcode는 간단한 C/C++ 프로그래밍부터 OS X용 응용프로그램 개발, iOS용 앱 개발이 가능한 전문 IDE이다.

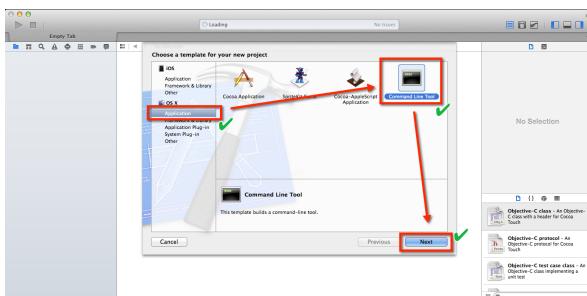
Mac OS X에 Xcode를 설치하고 간단히 실행하는 방법은 다음과 같다(Mac OS X의 앱스토어를 이용해 다운로드하고 설치한다.).



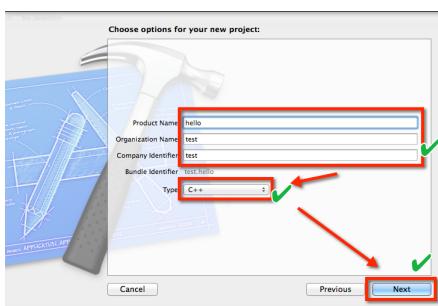
애플스토어를 통해 Xcode를 검색해 설치한다.



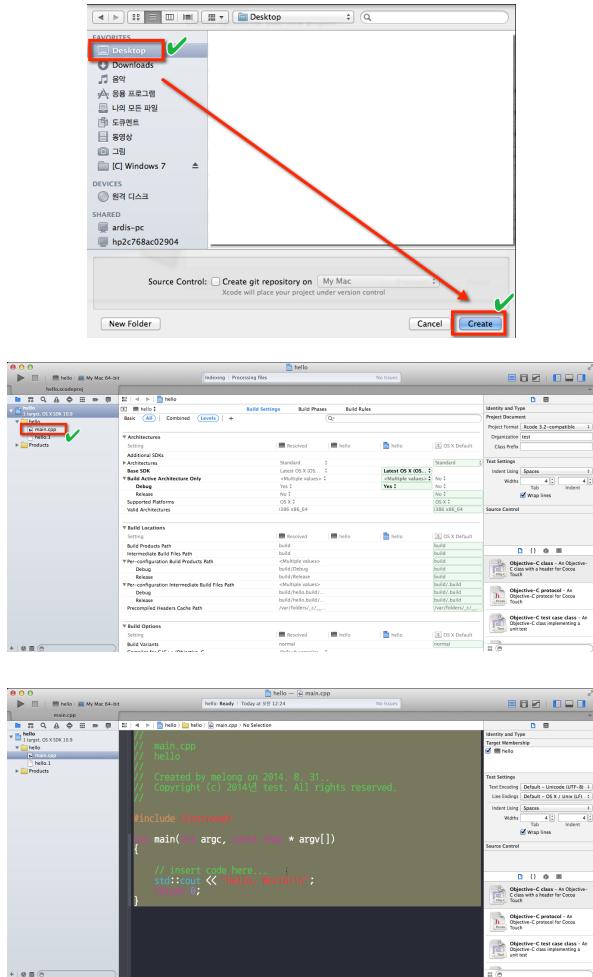
설치된 Xcode 실행 후
Create a new Xcode project
를 누른다.



OS X – Application을 선택하고
Command Line Tool을 선택한다.
Next를 누른다.



프로젝트 폴더 이름 등을 입력한다.
작성할 언어 타입을 C++로 지정한다.
Next를 누른다.



프로젝트 폴더를 만들 위치를 선택한다.

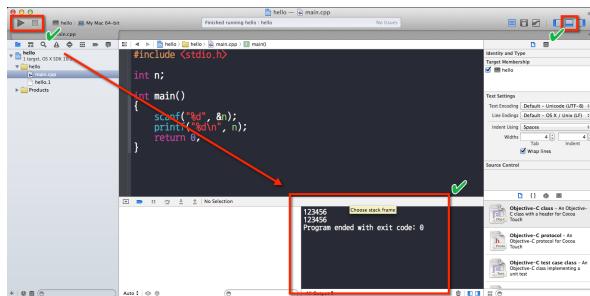
(Desktop은 바탕화면)

왼쪽 탐색창에서 `main.cpp`를 누른다.

소스 코드를 작성한다.

아래와 같은 기본 코드를 작성해 넣어보자

줄	코드(참고
1	#include <stdio.h>	
2		
3	int n;	
4		
5	int main()	
6	{	
7	scanf("%d", &n);	
8	printf("%d\n", n);	
9	return 0;	
10	}	



하단 창을 보이게 하고(오른쪽 위 아이콘에서 선택)

실행/정지 버튼을 누른다(왼쪽 위 아이콘에서 선택).

입력 출력이 하단 창에서 이루어진다.

라. 파일입출력 기본

채점을 위한 입출력이 파일로 설정되는 경우에는 지정된 파일로 데이터를 읽어 들어거나, 결과를 작성해야 한다. 아래는 가장 기본적인 입출력 예시이다.

- 표준 입출력

줄	코드	참고
1	#include <stdio.h>	7: 형식으로 입력
2		8: 형식에 맞추어
3	int n;	출력
4		
5	int main()	
6	{	
7	scanf("%d", &n);	
8	printf("%d\n", n);	
9	return 0;	
10	}	

- `freopen()` 표준입출력 재할당(변환)

줄	코드	참고
1	<code>#include <stdio.h></code>	
2		
3	<code>int n;</code>	
4		
5	<code>int main()</code>	
6	<code>{</code>	
7	<code> freopen("in.txt", "r", stdin);</code>	7: 표준입력(콘솔입력)의 내용을 in.txt 파일에서 읽어오도록 재할당
8	<code> freopen("out.txt", "w", stdout);</code>	8: 표준출력(콘솔출력)의 내용을 out.txt 파일에 기록하도록 재할당
9	<code> scanf("%d", &n);</code>	
10	<code> printf("%d\n", n);</code>	9: 형식으로 입력 10: 형식에 맞추어 출력
11	<code> return 0;</code>	
12	<code>}</code>	

- `fopen()` 파일처리

줄	코드	참고
1	<code>#include <stdio.h></code>	
2		
3	<code>int n;</code>	
4		
5	<code>int main()</code>	
6	<code>{</code>	
7	<code>FILE *in=fopen("in.txt", "r");</code>	7: in.txt 파일에서 데이터를 읽어오도록 지정
8	<code>FILE *out=fopen("out.txt", "w");</code>	8: out.txt 파일로 데이터를 출력하도록 지정
9	<code> fscanf(in, "%d", &n);</code>	9: 파일에서 형식으로 입력
10	<code> fprintf(out, "%d\n", n);</code>	10: 파일로 형식에 맞추어 출력
11	<code> return 0;</code>	
12	<code>}</code>	

실행파일 생성이나 파일 입출력은 운영체제의 사용자 권한과 관련한 문제들이 발생할 수 있기 때문에 때에 따라서 소스코드를 작성하고 실행시키기 위해서 추가적인 설정 작업이 필요한 경우도 있다.