

Coding Test

Academic Conference

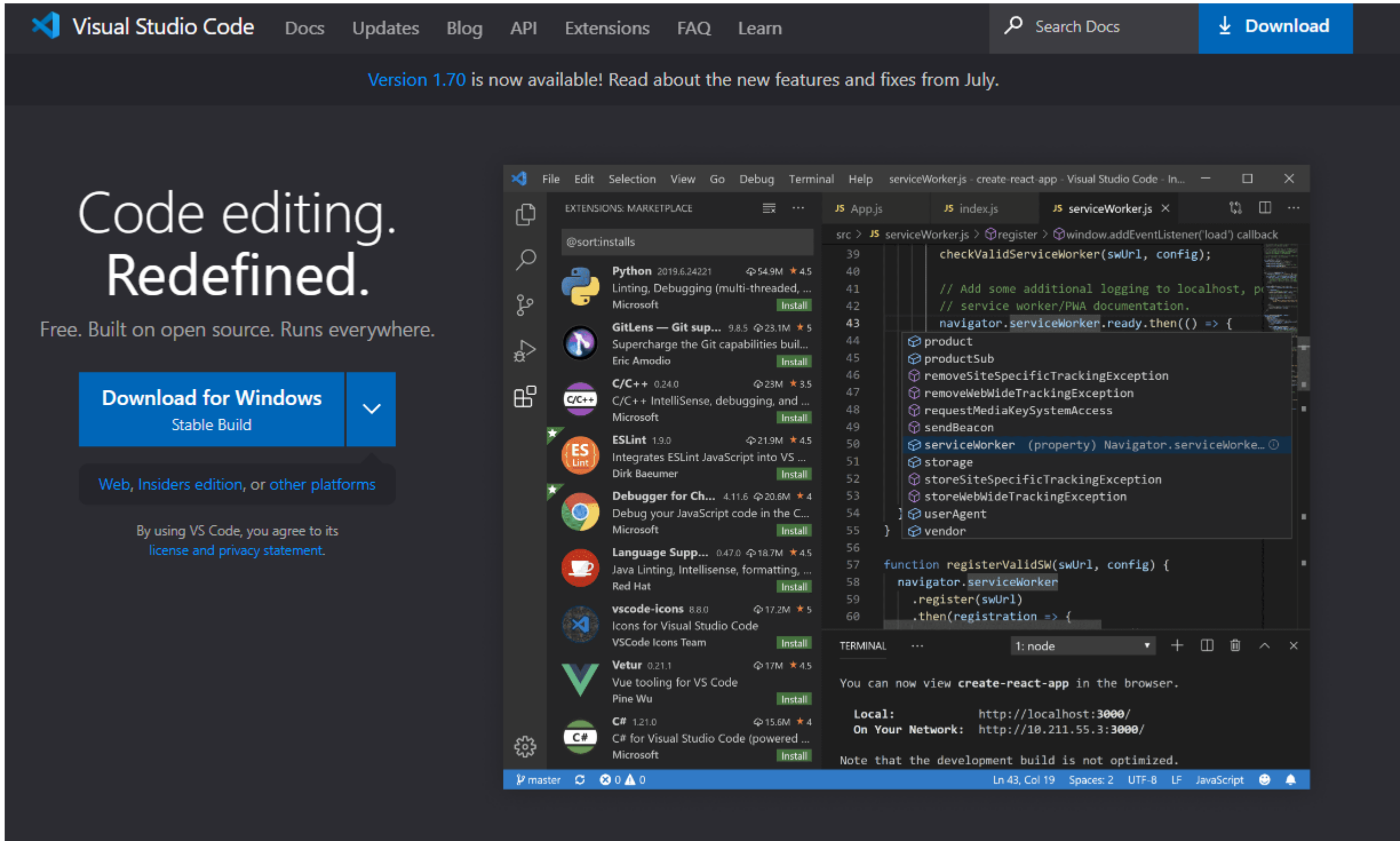


1. VSCode 환경 세팅
<https://code.visualstudio.com/>
2. 필수 C++
Type, 입/출력, 문자열, vector, Array, 포인터 등
3. 코테 필수 로직
Lower_bound, Upper_bound, 시계방향/반시계
방향 회전, 배열의 합, 배열 부분 회전 등
4. 코테 필수 수학
순열 등..

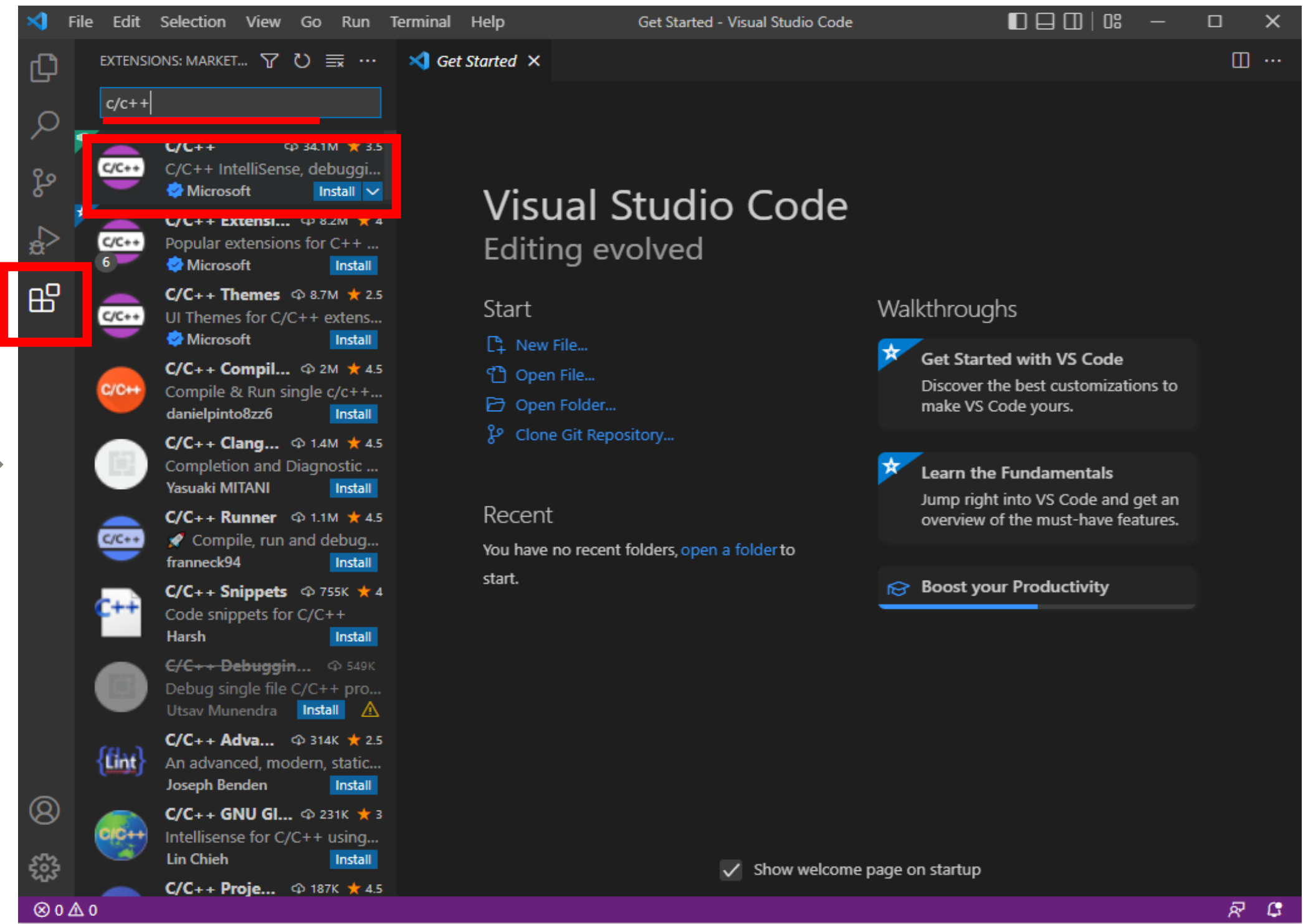
1. VSCode 환경 세팅

<https://code.visualstudio.com/>

1. visual studio code 자신의 버전에 맞춰 다운



2. Extensions에서 c/c++ 검색 후 install



1. VSCode 환경 세팅

https://sourceforge.net/projects/mingw/files/

3. MinGW 다운 - gcc compiler window에서 사용 가능하게끔

SOURCEFORGE

Open Source SoftwareBusiness SoftwareResources

창작 지원금 300만원과 어도비 협업 기회까지!

지금 확인하기

Home / Browse / Development / Build Tools / MinGW - Minimalist GNU for Windows / Files

MinGW - Minimalist GNU for Windows Files

A native Windows port of the GNU Compiler Collection (GCC)

Brought to you by: ctrauss, earnie, gressett, keithmarshall

SummaryFilesReviewsSupportNewsWikiMailing ListsTicketsGit

Download Latest Version

mingw-get-setup.exe (86.5 kB)

Get Updates

Name	Modified	Size	Downloads / Week
MinGW	2013-10-26		1,429,134
Installer	2013-10-04		1,518,807
Other	2011-11-13		1,888
MSYS	2011-11-13		1,634,748
README	2011-11-13	896 Bytes	22

Totals: 5 Items896 Bytes4,584,599

4. 빨간 표시 3개, 오른쪽 마우스 클릭>Mark for installation

MinGW Installation Manager

InstallationPackageSettings

Package	Class	Installed Version
mingw-developer-toolkit	bin	
mingw32-base	bin	
mingw32-gcc-ada	bin	
mingw32-gcc-fortran	bin	
mingw32-gcc-g++	bin	
mingw32-gcc-objc	bin	
msys-base	bin	

PackageClassInstalled VersionRepository VersionDescription

mingw-developer-toolkit	bin		2013072300	An MSYS Installation for MinGW Developers (
mingw32-base	bin		2013072200	A Basic MinGW Installation
mingw32-gcc-ada	bin		6,3,0-1	The GNU Ada Compiler
mingw32-gcc-fortran	bin		6,3,0-1	The GNU FORTRAN Compiler
mingw32-gcc-g++	bin		6,3,0-1	The GNU C++ Compiler
mingw32-gcc-objc	bin		6,3,0-1	The GNU Objective-C Compiler
msys-base	bin		2013072300	A Basic MSYS Installation (meta)

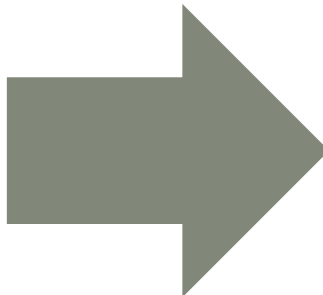
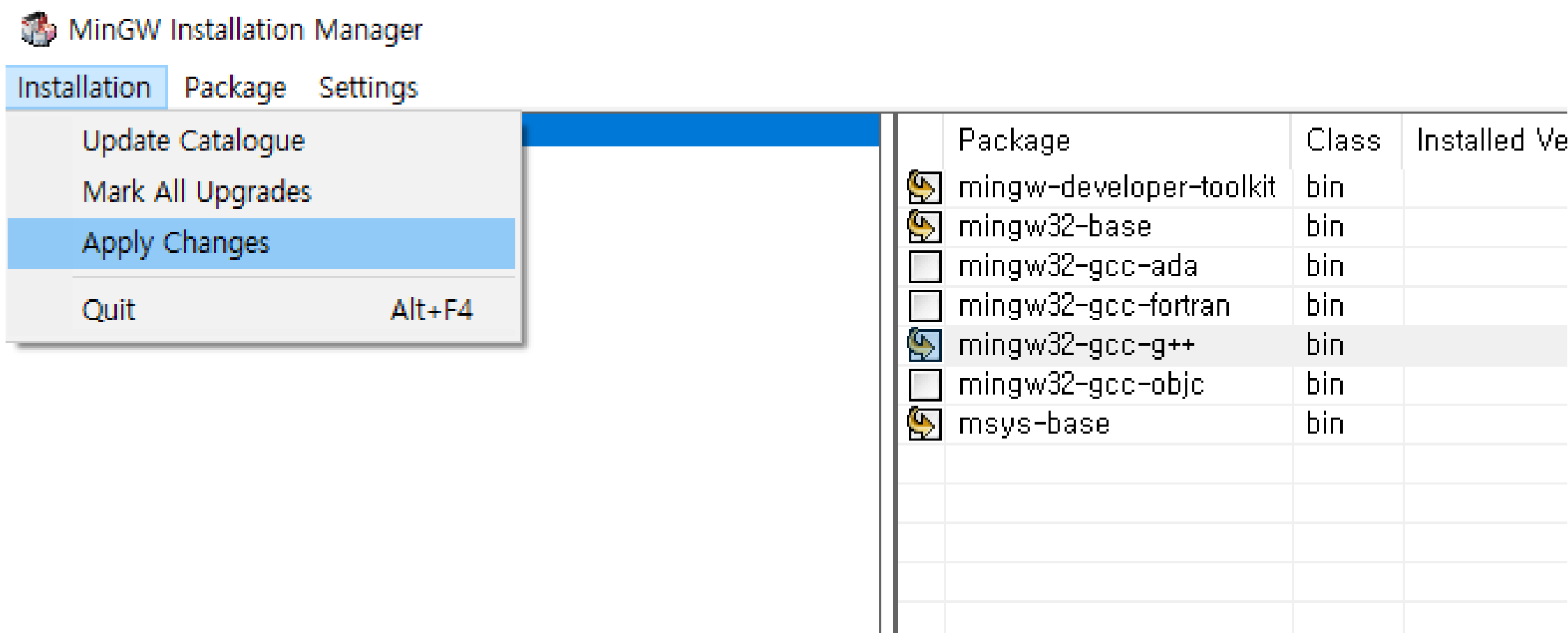
4번 result

Package	Class	Installed Version	Repository Version	Description
mingw-developer-toolkit	bin		2013072300	An MSYS Installation for M
mingw32-base	bin		2013072200	A Basic MinGW Installation
mingw32-gcc-ada	bin		6,3,0-1	The GNU Ada Compiler
mingw32-gcc-fortran	bin		6,3,0-1	The GNU FORTRAN Comp
mingw32-gcc-g++	bin		6,3,0-1	The GNU C++ Compiler
mingw32-gcc-objc	bin		6,3,0-1	The GNU Objective-C Cor
msys-base	bin		2013072300	A Basic MSYS Installation

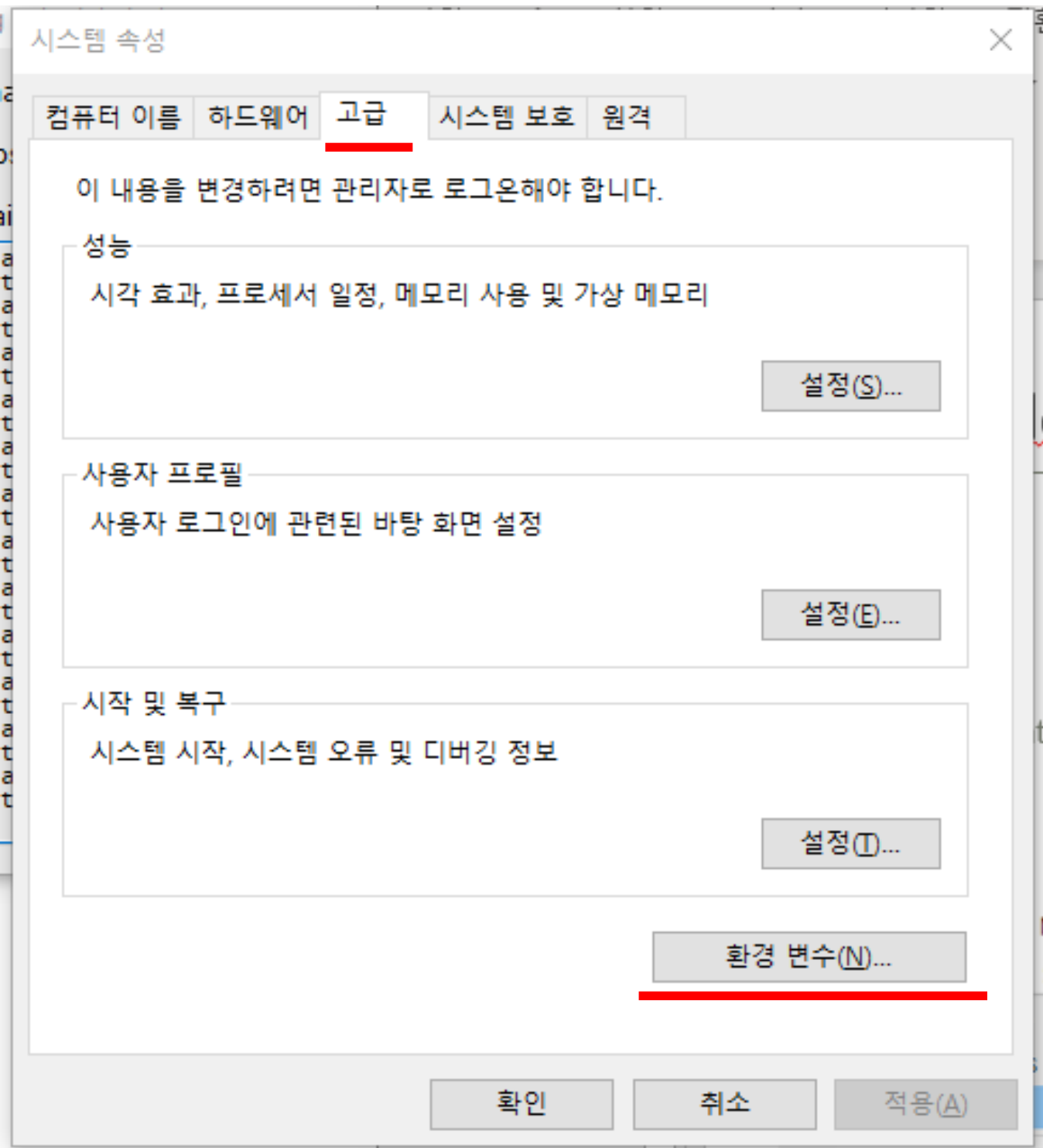
4

1. VSCode 환경 세팅

5. 왼쪽 상단 Installation>Apply Changes>Apply>close, 전체 창 close



6. 시스템 환경 변수 편집>고급>환경변수



1. VSCode 환경 세팅

7. 시스템 변수>Path 선택>편집>새로 만들기>MinGW 경로 추가 ex)C:\MinGW\bin
환경 변수

user에 대한 사용자 변수(U)

변수	값
OneDrive	C:\Users\User\OneDrive
OneDriveConsumer	C:\Users\User\OneDrive
Path	C:\Users\User\AppData\Local\Programs...
PyCharm Commu...	C:\Program Files\JetBrains\PyCharm Co...

새로 만들기(N)...

편집(E)...

삭제(D)

시스템 변수(S)

변수	값
NUMBER_OF_PRO...	16
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\W...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.W...

새로 만들기(W)...

편집(I)...

삭제(L)

확인

취소

8. 창 닫고 vs code재실행 후 hello.cpp 생성 및 코드 작성 (path인식 x 면 pc 재부팅)
확인 방법)cmd>gcc -v, g++ -v

9. 상단 메뉴 바 Terminal>Configure Default Build Task>c/c++ gcc.exe build active file 선택
어떤 것을 선택하든 tasks.json 나오면 됨

```
Get Started X G+ hello.cpp
C: > Users > user > G+ hello.cpp > main(void)
1 #include <iostream>
2 using namespace std;
3
4 int main(void){
5     cout<< "Hello World" <<endl;
6     return 0;
7 }
```

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS 1 unsaved

NO FOLDER OPENED

Run Task... Run Build Task... Run Active File Run Selected Text

Configure Tasks... Configure Default Build Task...

hello.cpp - Visual Studio Code

main(void)

<<endl;

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

.VS_CODE

tasks.json

argv.json

Select the task to be used as the default build task

C/C++: g++.exe 활성 파일 빌드

C/C++: cl.exe 활성 파일 빌드

save and compile for C++

save and compile for C

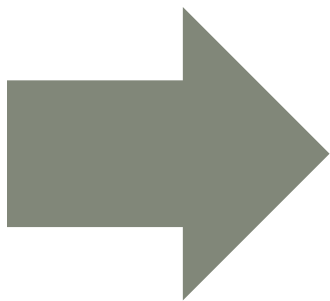
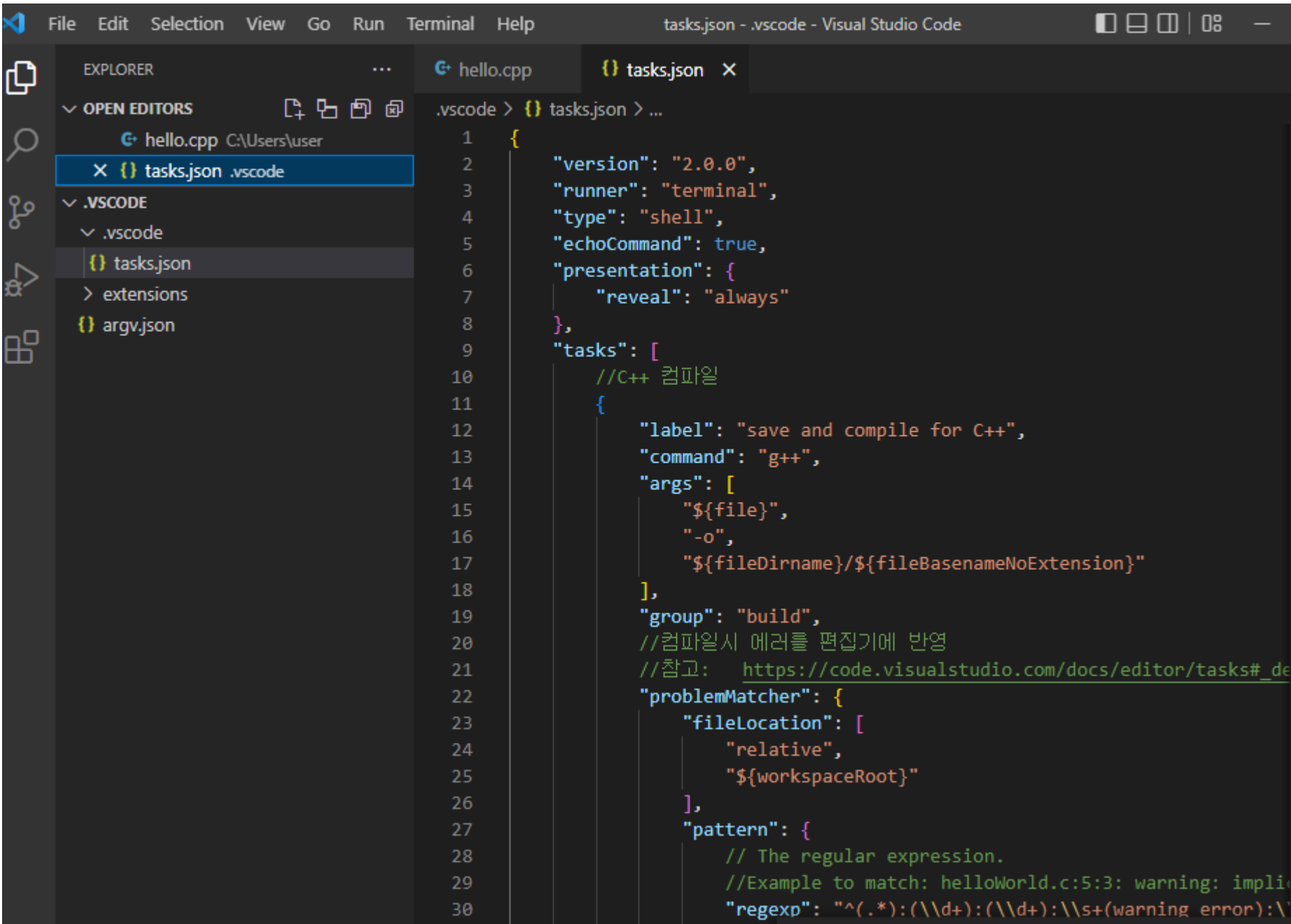
execute



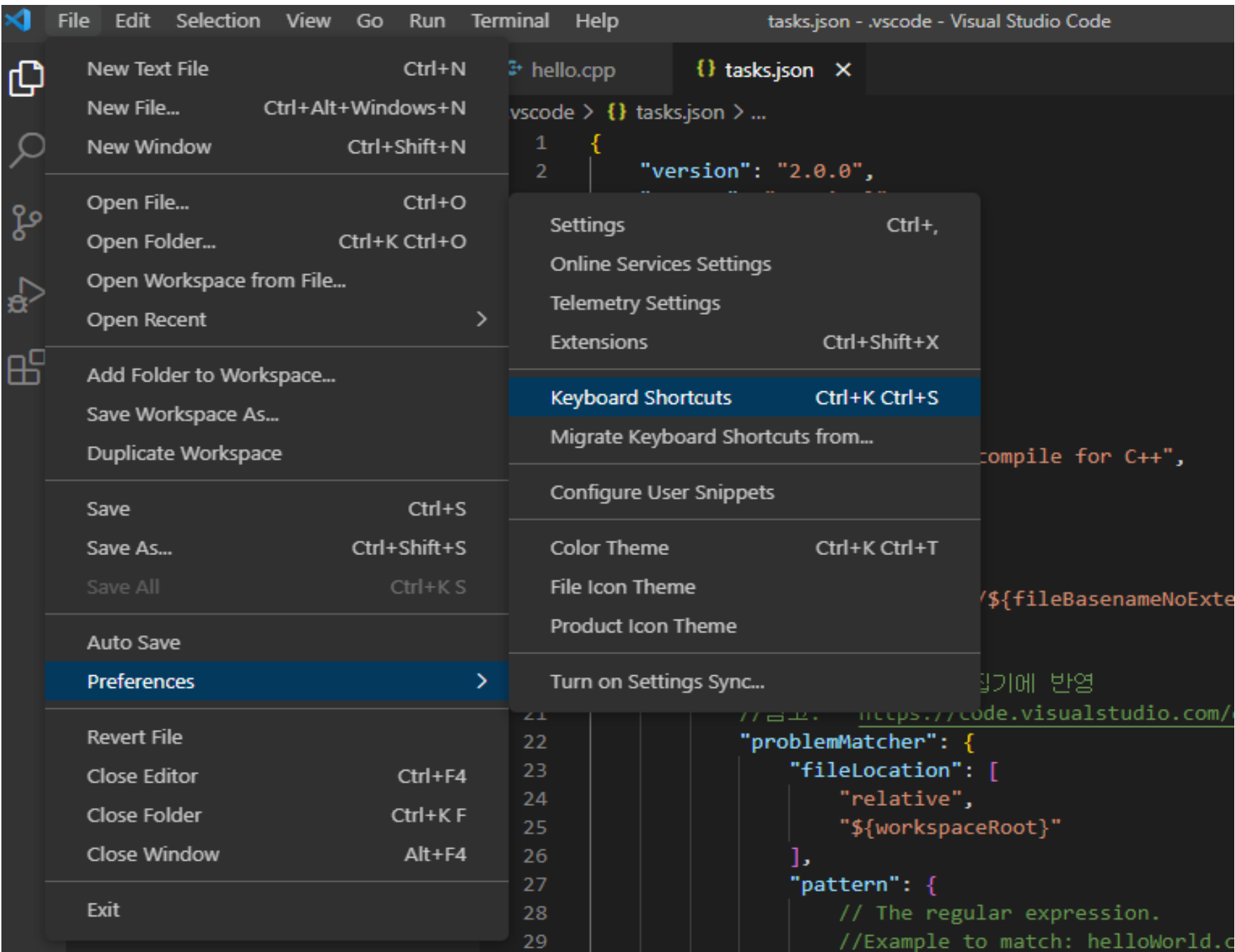
1. VSCode 환경 세팅

https://drive.google.com/file/d/1zSumWzSQBK4_JD7p8kGodjwcCjyD0Xqx/view?usp=sharing

10. tasks.json에 링크 내용 복붙하고 저장



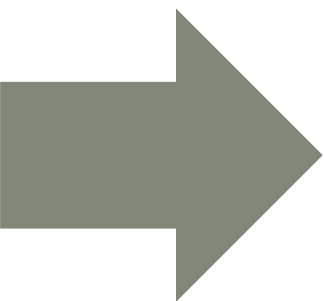
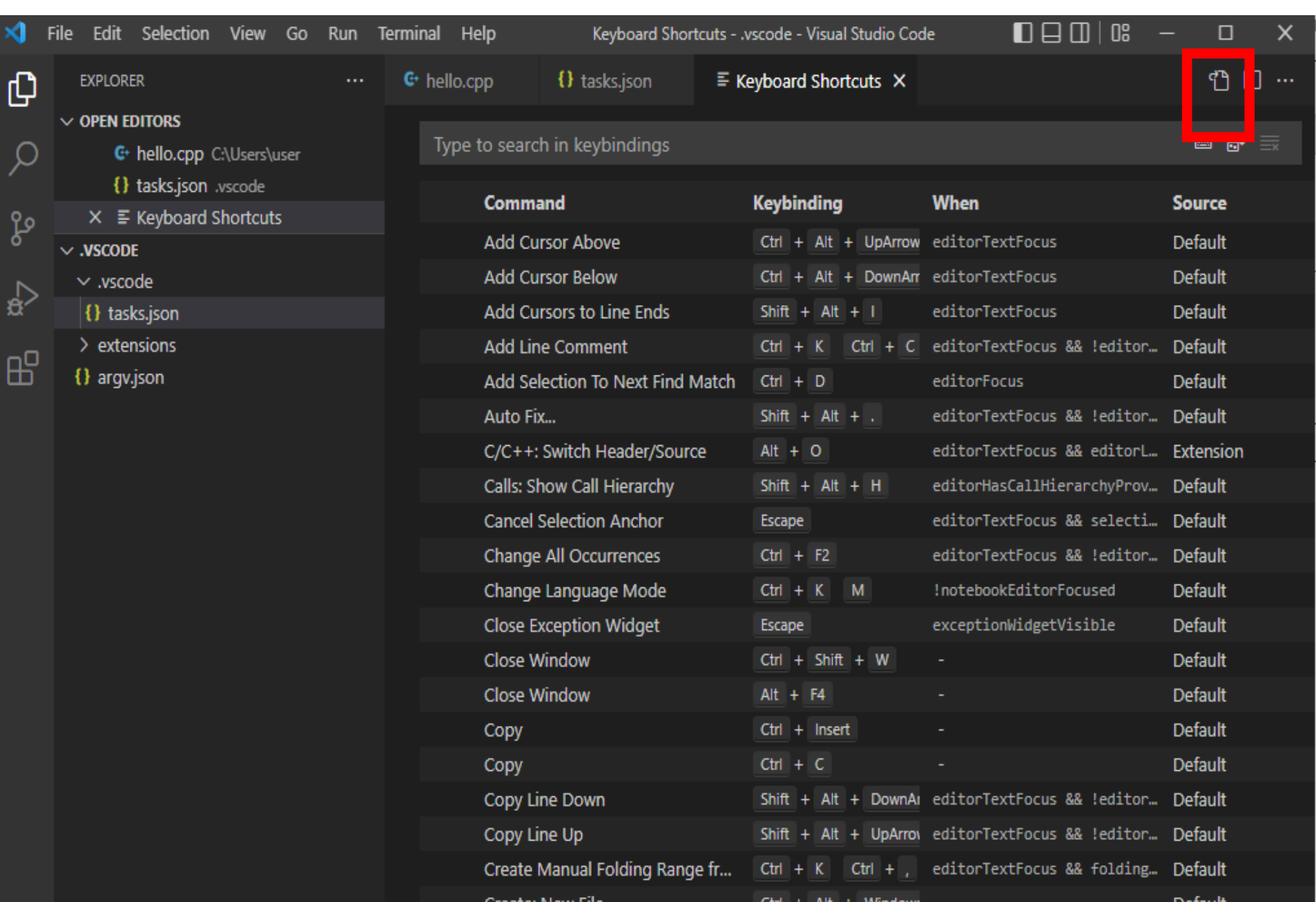
11. File>Preferences>Keyboard shortcuts



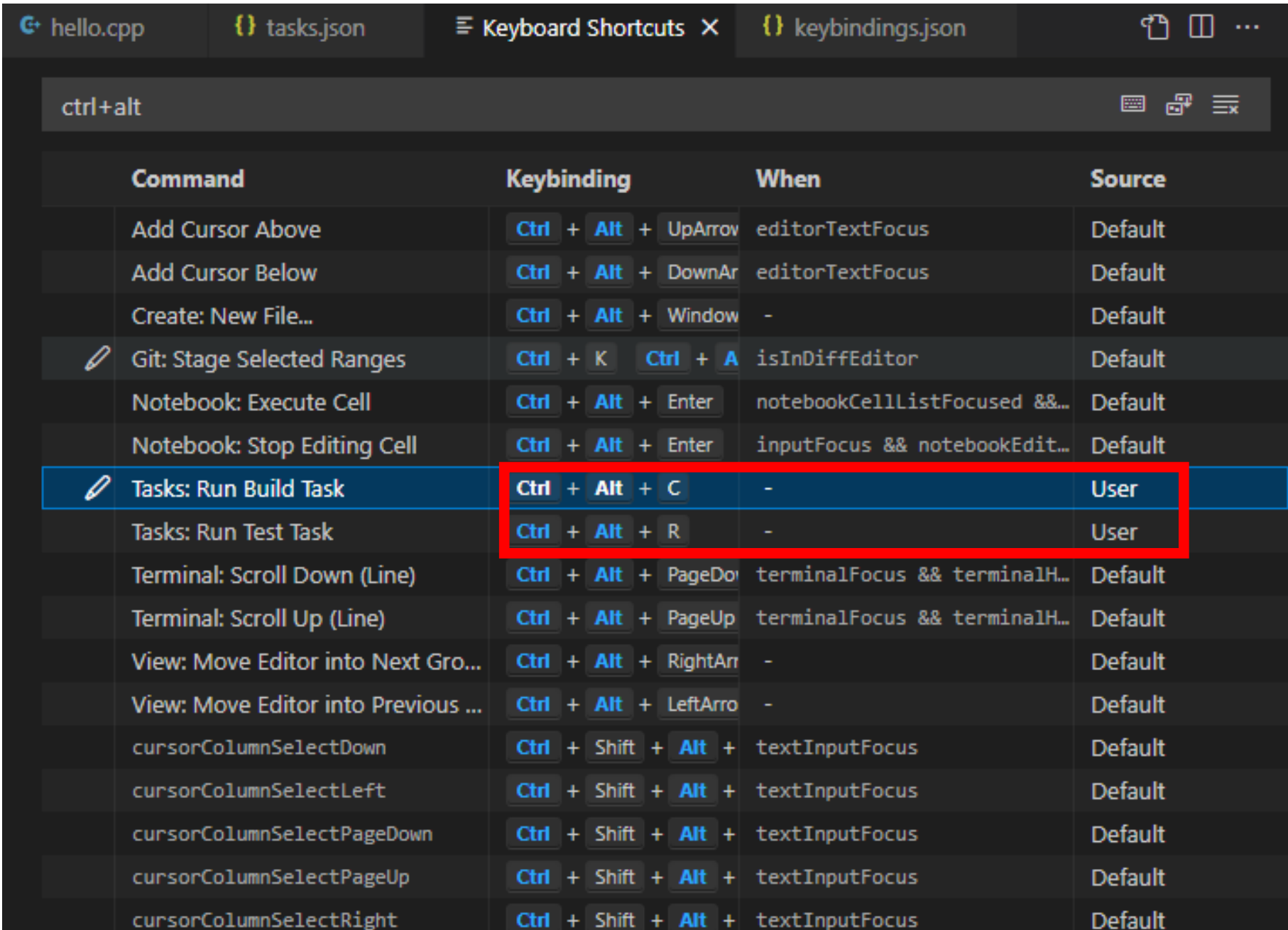
1. VSCode 환경 세팅

https://drive.google.com/file/d/11nowTUCnBfU_SKLzWn9GnYOLSSUJ7IKk/view?usp=sharing

12. 화면 우측 상단 Open Keyboard Shortcuts(JSON)>위 링크 내용 복붙 후 저장



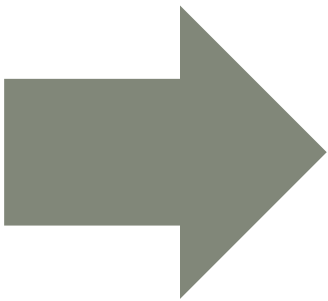
13. keyboard shortcuts에서 변경한 키 값이 User로 나오는 지 확인



1. VSCode 환경 세팅

14. ctrl+alt+c>enter 또는 팝업창 클릭으로 컴파일

```
• [icon] Executing task: gcc c:\Users\user\hello.cpp -o c:\Users\user/hello
```



15. ctrl+alt+r>enter 또는 클릭으로 실행

```
Hello World
[icon] Terminal will be reused by tasks, press any key to close it.
```

+) 수동으로 헤더파일 추가 하고 싶지 않을 때 하단의 헤더를 수동으로 다운받아 사용
#include <bits/stdc++.h>



2. 필수 c++ - type

```
#include <iostream>

using namespace std;

int main(){
    cout<<"Hello World!"<<"\n";
    return 0;
}
```

```
#include <iostream>

using namespace std;

void hello();

int main(){
    hello();
    return 0;
}

void hello(){
    cout<<"Hello!!"<<"\n";
}
```

```
#include <iostream>

using namespace std;

void hello(){
    cout<<"Hello!!"<<"\n";
}

int main(){
    hello();
    return 0;
}
```

type	byte	세부 설명
void		return 값 없음. 자료형이 없으므로 변수 만들 수 x
char	1	하나의 문자
string		문자로 이뤄진 배열. char[]보다 선호
bool	1	true, false or 0, 1
int	4	정수, -20억~+20억까지 표현 가능.
long long	8	Int로 표현할 수 없는 영역일 때 사용.
double	8	float보다 정밀함 필요시 사용.

- unsigned란?
-x ~ +x -> 0 ~ 2x
ex) unsigned long, unsigned double..

2. 필수 c++ - string, 아스키코드

```
//string
int main(){
    string a = "Hello World"; //char[12], 문자열의 끝을 알리는 \n 때문
    a += " ";
    a += "!";
    cout << a.size() << "\n"; //.size() 문자열 길이
    cout << a << "\n";
    cout << a[0] << "\n";
    cout << a[13] << "\n";
    cout << a[12] << "\n";
    return 0;
}
```

```
//아스키 코드와 문자열
//문자가 저장될 때 '3' 이 저장되는게 아니라 문자에 해당하는 아스키 코드 값
//이 저장되고 보여지는 것이기 때문에 형변환을 하지 않아도 숫자 덧셈이 가능
int main(){
    string s = "1234";
    s[2]++;
    cout << s << "\n";
    return 0;
}
```

Q. "\n" vs endl

2. 필수 c++ - 입력/출력

방법 1) cin, cout

형식이 정해지지 않은 경우

방법 2) scanf, printf

형식이 주어진 경우

```
#include <iostream>

using namespace std;

int a;
int b;
char c;
int main()
{
    // 정수 + 소수점 아래 자리 따로 받을 때
    scanf("%d.%d", &a, &b);
    scanf(" %c", &c);

    printf("십의 자리: %d, 일의 자리: %d\n", a, b);
    printf("a: %d, b: %d, c: %c", a, b, c);

    return 0;
}
```

3.1415

Y

십의 자리 : 3, 일의 자리 : 1415

a: 3, b: 1415, c: Y

2. 필수 c++ - 입력/출력

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	256 MB	87001	45024	40884	53.052%

문제

준하는 사이트에 회원가입을 하다가 joonas라는 아이디가 이미 존재하는 것을 보고 놀랐다. 준하는 놀람을 ??!로 표현한다. 준하가 가입하려고 하는 사이트에 이미 존재하는 아이디가 주어졌을 때, 놀람을 표현하는 프로그램을 작성하시오.

입력

첫째 줄에 준하가 가입하려고 하는 사이트에 이미 존재하는 아이디가 주어진다. 아이디는 알파벳 소문자로만 이루어져 있으며, 길이는 50자를 넘지 않는다.

출력

첫째 줄에 준하의 놀람을 출력한다. 놀람은 아이디 뒤에 ??!를 붙여서 나타낸다.

예제 입력 1 복사

```
joonas
```

예제 입력 2 복사

```
baekjoon
```

예제 출력 1 복사

```
joonas??!
```

예제 출력 2 복사

```
baekjoon??!
```

#10926

1. 알파벳, 50자 이내 아이디 입력받기
2. 아이디 뒤에 ??! 출력

```
#include <stdio.h>
int main(){
    char str[51];
    scanf("%s", str);
    printf("%s?\?!", str);
}
```

```
#include <iostream>
using namespace std;

int main(){
    char str[51];
    cin >> str;
    cout << str << "??!";
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main(){
    string str;
    cin >> str;
    str += "??!";
    cout << str;
    return 0;
}
```


2. 필수 c++ - type casting

되도록이면 형과 형을 똑같이 만들어주자!

ex) double/int = double, char-int = int

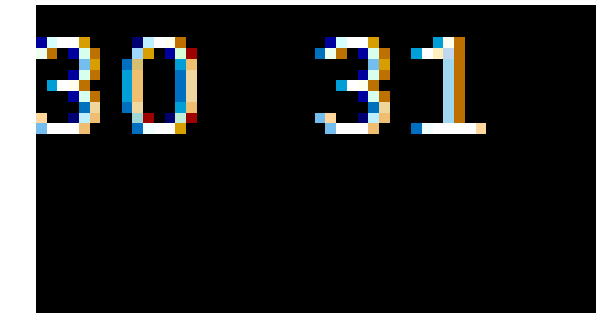
```
#include <iostream>

using namespace std;

float m = 3.14;
int main()
{
    int a = (int)m*10;
    int b = (int)10*m;

    printf("%d %d", a, b);

    return 0;
}
```



30 31

2. 필수 c++ -아스키코드

문제

알파벳 소문자, 대문자, 숫자 0-9중 하나가 주어졌을 때, 주어진 글자의 아스키 코드값을 출력하는 프로그램을 작성하시오.

입력

알파벳 소문자, 대문자, 숫자 0-9 중 하나가 첫째 줄에 주어진다.

출력

입력으로 주어진 글자의 아스키 코드 값을 출력한다.

예제 입력 1 복사

A

예제 입력 2 복사

C

예제 입력 3 복사

0

예제 출력 1 복사

65

예제 출력 2 복사

67

예제 출력 3 복사

48

Q. using namespace std ?

#11654

- 1. 알파벳 소/대문자, 숫자 0~9 중 하나 입력받음.
- 2. 입력값의 아스키 코드 출력

```
#include <stdio.h>
using namespace std;
char c;
int main(){
    scanf("%c", &c);
    printf("%d",c);
    return 0;
}
```

```
#include <iostream>
using namespace std;
char c;
int main(){
    cin >> c;
    cout << (int)c;
    return 0;
}
```

2. 필수 c++ - getline

띄어쓰기까지 받을 때,

```
#include <iostream>
using namespace std;
string s;

int main(){
    getline(cin, s);
    cout<<s;
}
```

```
#include<iostream>
using namespace std;
char s[30];

int main(){
    cin.getline(s, 30);
    cout<<s;
}
```

```
hello my name
hello my name
```

2. 필수 c++ - 문자열에서 자주 사용되는 함수

```
#include <iostream>

using namespace std;

string today = "today is mon";

int main(){
    cout << today.find("tue")<<"\n";
    cout << today.find(today[12]);
}
```

```
18446744073709551615
18446744073709551615
```

```
#include <iostream>
#include <algorithm>

using namespace std;

string today = "today is mon";

int main(){
    reverse(today.begin(), today.begin()+3);
    cout << today;
}
```

```
dotay is mon
```

string str = "Today is Monday"

-str.substr(0,3) //Tod

-str.find("찾을 문장") -> 없으면 string::npos, 있으면 문장 시작점 반환
중복된 값은 처음 등장한 위치 기준으로 반환
str.find("is")//6

-reverse(시작점, 끝) -> 시작점~끝-1 까지를 뒤집어주는 역할

2. 필수 c++ - 배열 vector 와 array

	Vector	Array
배열	동적 배열 ; 컴파일 시점에 개수 모를 때	정적 배열
형태	Vector<int> a	C스타일 int a[30] Std스타일 array<int, 30>
공통점	중복 허용하고 순서가 있음. 삽입, 삭제: O(1) , 랜덤 접근: O(n) 데이터 추가/삭제 많이 하는 것은 연결리스트, 탐색 많이 하는 것은 배열로!	

2. 필수 c++ - vector

Vector 초기화

vector<자료형> 변수명	벡터 생성
vector<자료형> 변수명(크기)	크기만큼 벡터 생성 후 0으로 초기화
vector<자료형> 변수명(크기, 초기화 값)	크기만큼 초기화 값으로 초기화
vector<자료형>변수명 = {변수1, 변수2...}	벡터 생성 후 오른쪽 변수 값으로 초기화
vector<자료형> 변수명[] = {{1,2,3},{4,5,6}}	벡터 배열(2차원 벡터) 선언 및 초기화(열은 고정, 행은 가변)
Vector<vector<자료형>>변수명	2차원 벡터 생성 (열과 행 모두 가변)
Vector<자료형>변수명.assign(범위, 초기화값)	벡터의 범위 내에서 해당 값으로 초기화

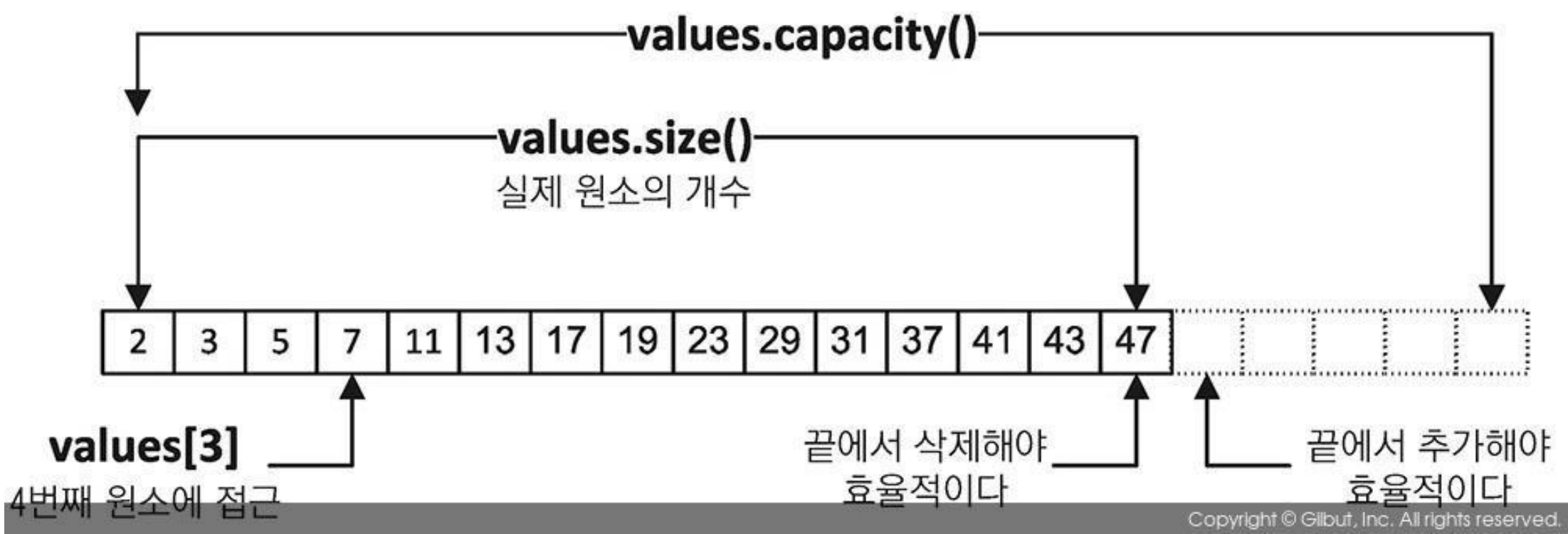
2. 필수 c++ - vector

```
#include<iostream>
#include<vector>
using namespace std;

vector<int> v;

int main(){
    for(int i=1; i<=5;i++)v.push_back(i);
    cout<<*v.begin()<<"\n";
    cout<<*(v.end()-1)<<"\n";
}
```

```
std::vector<size_t> values {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47};
values.reserve(20);           // 용량을 20으로 늘린다
```



Vector iterator

v.begin()	벡터 시작점의 주소 값 반환
v.end()	벡터 (끝부분 + 1) 주소값 반환
v.push_back(input 값)	뒤에 input 된 요소를 더함
v.pop_back()	맨 뒤부터 지움
emplace_back()	맨 뒤부터 지움, 더 빠름
v.erase(범위 또는 index 번호)	() 값 지우기
v.clear()	Size는 0, capacity는 그대로
v.rbegin()	벡터의 끝 지점을 시작점으로 반환
v.rend()	벡터의 시작+1 지점을 끝 부분으로 반환

2. 필수 c++ - vector

```
#include <iostream>
#include <vector>

using namespace std;

vector<int> v;
int main()
{
    for(int i = 1; i<=10; i++) v.push_back(i); //1~10
    for(int a : v) cout << a << " "; //int a : v ; 배열 v의 요소를 순차적으로 탐색한다는 의미
                                     //자료형 a : v로 pair의 경우 pair<int,int> a: v 로 사용
                                     //for(int i=0; i<v.size(); i++) v[i] 와 같은 의미
    cout << "\n";
}
```

<algorithm>

find(v.begin(), v.end(), num) // 원소를 찾으면 원소 위치의 iterator 반환, 존재하지 않으면 v.end()반환

2. 필수 c++ - Array

배열의 초기화

- fill(시작, 끝, 초기화 값), memset(초기화 하는 배열, 값, 배열의 크기) //바이트 단위
초기화로 0, -1 보통 하나의 문자로 초기화할 때 사용)

배열의 복사

어떤 배열을 수정할 때 원본 배열을 유지시키고 싶으면 사용
- memcpy(새롭게 넣을 배열, 복사원본 배열, 복사원본 배열의 크기)

2. 필수 c++ - Sort()

```
using namespace std;

vector<pair<int, int>> v;

bool cmp(pair<int, int> a, pair<int, int> b){
    return a.first > b.first;
}

int main(){
    for(int i = 10; i >= 1; i--){
        v.push_back({i, 10 - i});
    }

    sort(v.begin(), v.end(), cmp);
    for(auto it : v) cout << it.first << " : " << it.second << "\n";
    return 0;
}
```

sort(처음 이터레이터, 끝 이터레이터, *커스텀 비교 함수) // 최소 2개는 매개변수로 넣어야 함

-배열 등 컨테이너들의 요소를 정렬하는 함수

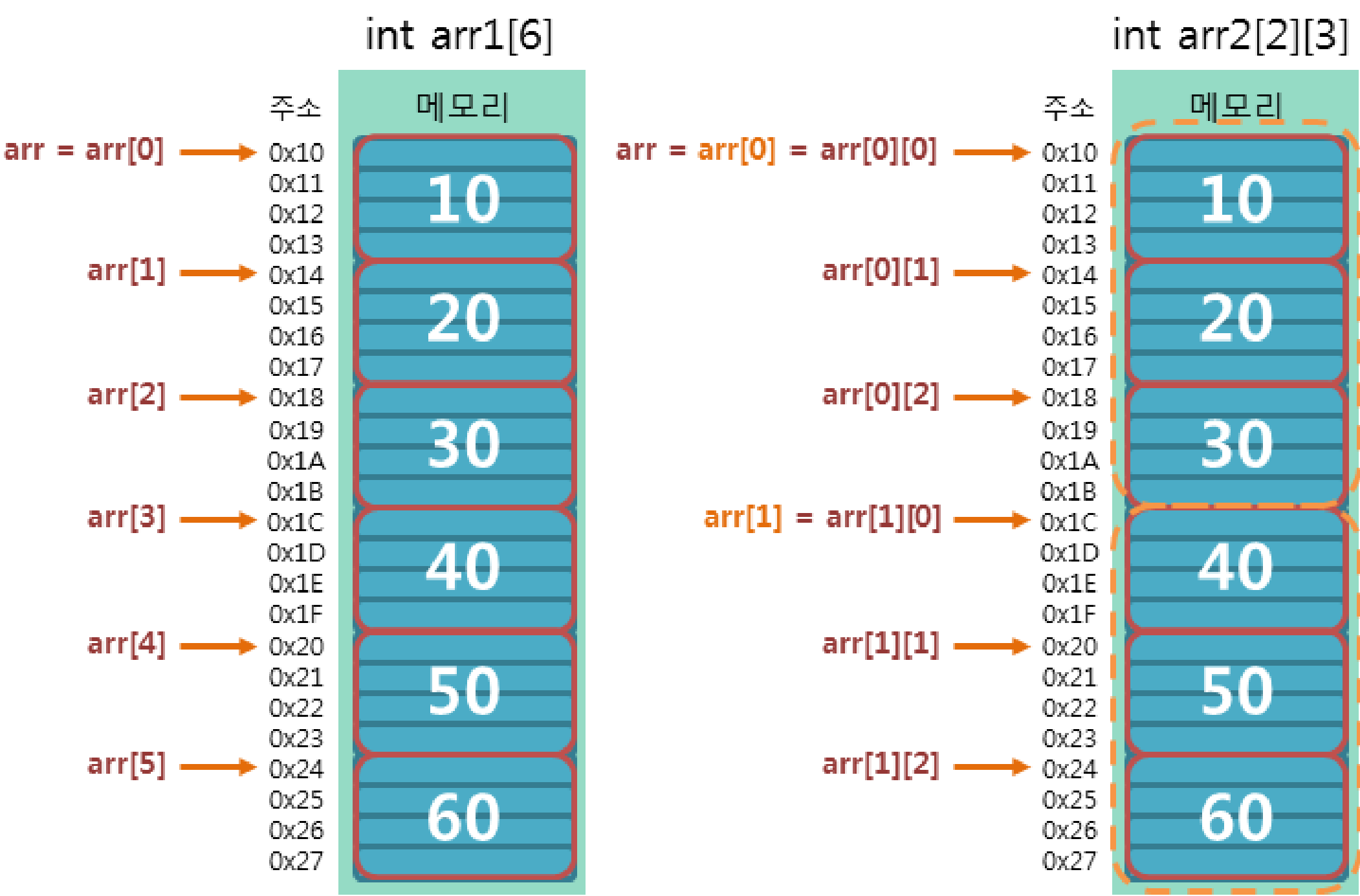
-범위 [first, last)

-커스텀 비교 함수를 넣지 않으면 기본적으로 오름차순 정렬

-greater<int>()를 넣어 내림차순 정렬 가능 // less<int>() 넣으면 오름차순

2. 필수 c++ - 2차원 배열

a[i][j] // i는 행, j는 열



탐색할 때는 순서를 지키면 효율성이 높아짐

```
using namespace std;

int a[y][x];

int main()
{
    // 첫번째 차원 > 두번째 차원 순으로 탐색하면 좋음
    // c++에서 캐시를 첫번째 차원(y)을 기준으로 하는데, 캐시관련 효율성을 생각하면
    // 탐색에도 순서를 지켜주면 좋음
    for(int i = 0; i < y; i++){
        for(int j = 0; j < x; j++){
            cout << a[i][j] << " ";
        }
        cout << "\n";
    }

    for(int i = 0; i < x; i++){
        for(int j = 0; j < y; j++){
            cout << a[j][i] << " ";
        }
        cout << "\n";
    }
}
```

2. 필수 c++ - split

<sstream> <https://chbuljumeok1997.tistory.com/42>

```
#include<iostream>
#include<vector>
#include<sstream>
using namespace std;

vector<string> v;

int main(){
    string str = "Hello World";
    istringstream ss(str);
    cout<<ss.str()<<"\n"; //Hello World
    string n;
    while(ss>>n){
        // cout<<n<<"\n";
        v.push_back(n);
    }
    cout<<v[0];
    return 0;
}
```

```
#include<iostream>
#include<vector>
#include<sstream>
#include<algorithm>
using namespace std;

int main(){
    string str = "Hello World";
    istringstream ss(str); //ss.str() 해보면 Hello world 출력
    string stringBuf;
    vector<string>v;
    v.clear(); //초기화

    //getline(input, buffer, delim) input이란 istream& input 을 말함.
    while(getline(ss,stringBuf, ' ')){
        v.push_back(stringBuf);
    }
    for(string i : v)cout<<i<<"\n";
}
```

2. 필수 c++ - split

split 구현

substr(시작, 끝); 문자열 부분 return

```
using namespace std;

vector<string> split(string input, string delimiter) {
    vector<string> ret;
    long long pos = 0;
    string token = "";
    while ((pos = input.find(delimiter)) != string::npos) {
        token = input.substr(0, pos);
        ret.push_back(token);
        input.erase(0, pos + delimiter.length());
    }
    ret.push_back(input);
    return ret;
}

int main(){
    string s = "Hello My name is C++";
    string d = " ";
    vector<string> a = split(s, d);
    for(string b : a) cout << b << "\n";
}
```

2. 필수 c++ - split

단어의 개수

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	128 MB	253223	77273	61628	31.582%

문제

영어 대소문자와 공백으로 이루어진 문자열이 주어진다. 이 문자열에는 몇 개의 단어가 있을까? 이를 구하는 프로그램을 작성하시오. 단, 한 단어가 여러 번 등장하면 등장한 횟수만큼 모두 세어야 한다.

입력

첫 줄에 영어 대소문자와 공백으로 이루어진 문자열이 주어진다. 이 문자열의 길이는 1,000,000을 넘지 않는다. 단어는 공백 한 개로 구분되며, 공백이 연속해서 나오는 경우는 없다. 또한 문자열은 공백으로 시작하거나 끝날 수 있다.

출력

첫째 줄에 단어의 개수를 출력한다.

예제 입력 1 복사

```
The Curious Case of Benjamin Button
```

예제 입력 2 복사

```
The first character is a blank
```

예제 입력 3 복사

예제 출력 1 복사

```
6
```

예제 출력 2 복사

```
6
```

예제 출력 3 복사

#1152

- 1. 대소문자와 공백으로 이뤄진 문자열 입력
- 2. 단어 중복된건 각각 count
- 3. 문자열 길이는 10,000,000 넘지 x
- 4. 문자열 공백으로 시작하거나 끝날 수 있음, 공백 연속 x

고려해야할 부분! 공백으로 시작, 끝나는 단어 |

```
#include <iostream>
#include <string>

#define MAX 1000001 //str.length로 고려해줄 수도 있음

using namespace std;

int main(){
    string str;
    int cnt = 1;
    getline(cin, str);

    for(int i = 0; i<str.length(); i++){
        if(str[i] == ' ') cnt++;
    }

    if(str[0] == ' ')cnt--; //처음 공백이 올 때
    if(str[str.length()-1] == ' ')cnt--;//맨 뒤가 공백
    cout << cnt;
    return 0;
}
```

3. 코테 자주 등장하는 logic - accumulate

배열의 합을 쉽고 빠르게 구해주는 함수!

```
#include<iostream>
#include<vector>
#include<numeric>

using namespace std;

int main(){
    vector<int> v = {1,2,3,4,5};
    int sum = accumulate(v.begin(), v.end(), 0);
    cout << sum << "\n";
}
```


3. 코테 자주 등장하는 logic – *max_element, *min_element

배열 중 가장 큰 요소/작은 요소 추출하는 함수!

```
#include<iostream>
#include<vector>
#include<algorithm>

using namespace std;

int main(){
    vector<int> v = {1,2,3,4,5};
    int a = *max_element(v.begin(), v.end());
    int b = *min_element(v.begin(), v.end());

    cout<<a<<'\\n';
    cout<<b<<'\\n';
}
```

3. 코테 자주 등장하는 logic – 배열 부분 이동(회전)

1번부터 3번까지의 배열만 앞으로 한칸씩 이동된 로직을 구현하기 위해서는?

```
#include<iostream>
#include<vector>

using namespace std;

vector<int> v;

int main(){
    for(int i = 1; i<=5; i++) v.push_back(i);
    int i = 1;
    int temp = v[i];
    v[i] = v[i+1];
    v[i+1] = v[i+2];
    v[i+2] = temp;
    for (int i : v) cout<< i << ' ';
    return 0;
}
```

1 3 4 2 5

3. 코테 자주 등장하는 logic – n진법 변환

<많이 사용되는 진법>

-2진법: 0~1까지 2개의 숫자로 수를 표현

-8진법: 0~7까지 8개의 숫자로 수를 표현

-16진법: 0~9까지의 10개의 숫자와 남은 자리는 A(65)~F까지 6개의 문자를 사용해 수를 표현

<10진수 a -> n 진법 표기 원리>

1. $a \% n$ 을 배열에 순서대로 넣는다.
2. $a = a / n$ 을 하고 1번을 반복한다.
3. 단, a가 0일때 종료되며, a가 1일때는 배열에 1을 넣어주고 종료한다.
4. 배열을 뒤집어서 출력해준다.
5. 단, 16진법을 고려하여 배열에 숫자 10이상이 등장하면 +55를 해 A~F 로 바꿔준다.

```
#include<iostream>
#include<vector>
#include<algorithm>

using namespace std;

vector<int> v;
int main(){
    int a = 100; //10진수
    int n = 2; //바꿀 진법의 n
    while(a>1){
        v.push_back(a%n);
        a/=n;
        if(a==1)v.push_back(1);
    }
    reverse(v.begin(), v.end());
    for(int i:v){
        if(i>=10)cout << char(i+55);
        else cout<<i;
    }
    return 0;
}
```

3. 코테 자주 등장하는 logic – 내림차순 정렬

Sort 세 번째 인자에 커스텀 함수(대체적으로 bool형 선언) 넣는 경우

: 대부분 vector 가 담고 있는 자료형이 int가 아닌 경우 많이 사용. ex) vector<pair<int, int>>v;

sort 함수에서 세 번째 인자로 들어간 함수를 인식 -> 자동으로 앞에 수를 i, 뒤에 수를 j에 넣어주고, i > j가 참이면 1, 거짓이면 0을 반환하는 함수이므로, 참일때 sort를 실시하게 되어 내림차순으로 정렬된다.

```
#include<iostream>
#include<vector>
#include<algorithm>

using namespace std;

vector<int> v;

bool cmp(int a, int b){
    return a>b;
}

int main(){
    for(int i = 1; i<=5; i++)v.push_back(i);
    sort(v.begin(),v.end(),cmp);//sort(시작, 끝, greater<int>() or x) -> 오름차순
    //sort(시작, 끝, less<int>() or custom 함수) -> 내림차순
    for(int i : v)cout<<i<<' ';
}
```

3. 수학 – 등차수열의 합, 거듭제곱, 세제곱 합공식

등차수열

:연속한 두 항의 차를 d라 할 때, d의 값이 일정한 수열

$$a_{k+1} - a_k = d$$

$$\sum_{k=1}^n k = \frac{n(n+1)}{2} \quad : 1씩 증가하는 수열의 합 공식$$

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6} \quad : 1, 4, 9..., 거듭제곱 합 공식$$

$$\sum_{k=1}^n k^3 = \left\{ \frac{n(n+1)}{2} \right\}^2 \quad : 1, 8, 27..., 세제곱 합 공식$$

3. 수학 – 최대 공약수와 최소 공배수

<https://www.acmicpc.net/problem/2609>

최대 공약수

: 0이 아닌 2개 이상의 정수의 공통된 약수 중 가장 큰 수

0과 n의 최대공약수는 n

최소 공배수

: 두 개 이상의 자연수의 공통인 배수(공배수) 중 가장 작은 수

소인수 분해하여 각 수 중에서 적어도 한 곳에라도 있는 약수를 모두 찾아서 곱해주면 됨

소인수

: 약수 중 소수인 약수로, 소수는 1과 자기 자신만을 약수로 가지는 수를 말함

```
int gcd(int a,int b){
    //a>=b 라 하면
    if(b==0)return a;
    return gcd(b, a%b);
}

int lcm(int a,int b){
    return (a*b)/gcd(a,b);
}

int main(){
    cout<<gcd(60,66);
}
```

3. 수학 – 모듈러 연산

<https://www.acmicpc.net/problem/16428>

모듈러 연산

: 정수론에서 모듈러 연산이란 정수의 합과 곱을 어떤 주어진 수의 나머지에 대하여 정의하는 방법을 말함

Ex)

$$17 \bmod 5 = 2$$

$$7 \bmod 10 = 7$$

모듈러 합동

$$(a \bmod n) = (b \bmod n) \rightarrow a \equiv b \pmod{n}$$

모듈러 연산의 속성 <https://torbjorn.tistory.com/228>

$$(a + b) \% M = ((a \% M) + (b \% M)) \% M$$

$$(a - b) \% M = ((a \% M) - (b \% M) + M) \% M$$

$$(a \times b) \% M = ((a \% M) \times (b \% M)) \% M$$

뱀셈의 경우, mod 연산을 한 결과가 음수가 나올 수 있기 때문에 M 을 더해줘야 합니다.

3. 수학 – 에라토스테네스 == 소수 판별 문제

<https://www.youtube.com/watch?v=5ypkoEgFdH8>

에라토스테네스의 채

: 소수를 찾는 방법으로 bool 배열을 선언하여 주로 사용
굉장히 많은 식 있음

- 1. 2는 소수이므로 그대로 두고 n까지 2의 배수들을 0으로 치환
- 2. 3은 소수이므로 그대로 두고 n까지 3의 배수들 0으로 치환
- 3. 4는 2번에서 0으로 치환 됐으므로 pass
- 4. 위의 과정을 n의 제곱근에서 내림한 수까지 반복
ex) $11^2 > 120$ 이므로 11보다 작은 수의 배수들만 체크
- 5. 2 부터 n까지 0으로 치환되지 않은 수 출력

```
using namespace std;

bool check(int n) {
    if(n <= 1) return 0;
    if(n == 2) return 1;
    if(n % 2 == 0) return 0;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) return 0;
    }
    return 1;
}
```

간단히
N<=1 : 소수x
N==2 : 소수
n%2 == 0 : 소수x
그 외의 수 i++ 해가며 n을 나누고 %나머지 확인

```
#include<iostream>
#include<cmath>
using namespace std;

int n = 100; // 구하고자 하는 숫자 범위
int prim_n[101];

void eratos(){
    //prim_n 초기화
    for(int i = 2; i<=n; i++){
        prim_n[i] = i;
    }
    for(int i = 2; i <= sqrt(n); i++){
        if(prim_n[i] == 0)continue;
        for(int j = i * i; j<=n; j += i)prim_n[j] = 0;
    }
    for(int i=2; i<=n; i++){
        if(prim_n[i] != 0)
            printf("%d\n", prim_n[i]);
    }
}

int main(){
    eratos();
}
```

```
vector<int> era(int mx){
    vector<int> v;
    vector<int> tmp;//max number(mx) 만큼 숫자 배열 생성
    for(int i = 2; i <= mx; i++){
        if(tmp[i]) continue;
        for(int j = 2*i; j <= mx; j += i){
            tmp[j] = 1;
        }
    }
    for(int i = 2; i <= mx; i++) if(tmp[i] == 0)v.push_back(i);
    return v;
}
```

+ Tip
