Quick Sublime Install Reference

1) Download sublime here.
    a) Windows users: double click on the inteller
    b) Mac users: double click on dmg and create the **symbolic link by performing the following command

    ln -s "/Applications/Sublime Text.app/Contents/SharedSupport/bin/subl" /usr/local/bin/subl

    Note: Symbolic links or symlinks are files that simply contain the path of the target file or directory stored as text. If you move the target file, the symlink will break because it still points to the original location in the link. Thus, while using the link in the command line we will have to use a shorter and simpler link.

    c) Linux users:
        i) Download and uncompress it manually or
        ii) Perform the following commands:

    $ cd ~
    $ wget http://c758482.r82.cf2.rackcdn.com/sublime-text_build-3083_amd64.deb
    $ cd ~
    $ wget http://c758482.r82.cf2.rackcdn.com/sublime_text_3_build_3083_x64.tar.bz2
    $ sudo mv Sublime\ Text\ 3 /opt/

2) Configure sublime.
    a) Install PackageResourceViewer plugin
        i) Installation through package control is recommended. It will handle updating your packages as they become available.
        ii) The console is accessed via the *ctrl+`* shortcut or the View > Show Console menu. Once open, paste the appropriate Python code for your version of Sublime Text into the console from this link.
        iii) To install go to the *Command Palette* and enter *Package Control: Install Package*
        iv) Search for *PackageResourceViewe*r.

    b) Open the command center by entering *Ctrl + Shift + p*
    c) Type "*PackageResourceviewer: Open Resource*"
    d) Then *C++ –> C++ Single File.Sublime-build* , this will open the build config file for C++
    e) Next to every instance of *g++* add *-std=c++11* in front of it and save the file.

```
{
	"shell_cmd": "g++ -std=c++11 \"${file}\" -o \"${file_path}/${file_base_name}\"",
	"file_regex": "^(..[^:]*):([0-9]+):?([0-9]+)?:? (.*)$",
	"working_dir": "${file_path}",
	"selector": "source.c, source.c++",
	"variants":
	[
		{
			"name": "Run",
			"shell_cmd": "g++ -std=c++11 \"${file}\" -o \"${file_path}/${file_base_name}\"
&& \"${file_path}/${file_base_name}\""
		}
	]
}
```

f) Save it. Rename the file as “MyC++Build” or anything you want and save it.
g) Go into *Tools > Build System* and then Select your Build “*MyC++Build*”.
h) Later open any folder using *Ctrl+O* where you usually store your programs. In that folder create a file name “*input.txt*”.

Now start coding. Write a program and then save it in that folder only. The benefit of doing is that whenever you have to give inputs just open the created *input.txt* enter your input data and save it. Later perform *Ctrl+B* for compiling and *Ctrl+Shift+B* for Run.