VScode Setup

If you dont have GCC installed, download it from here:

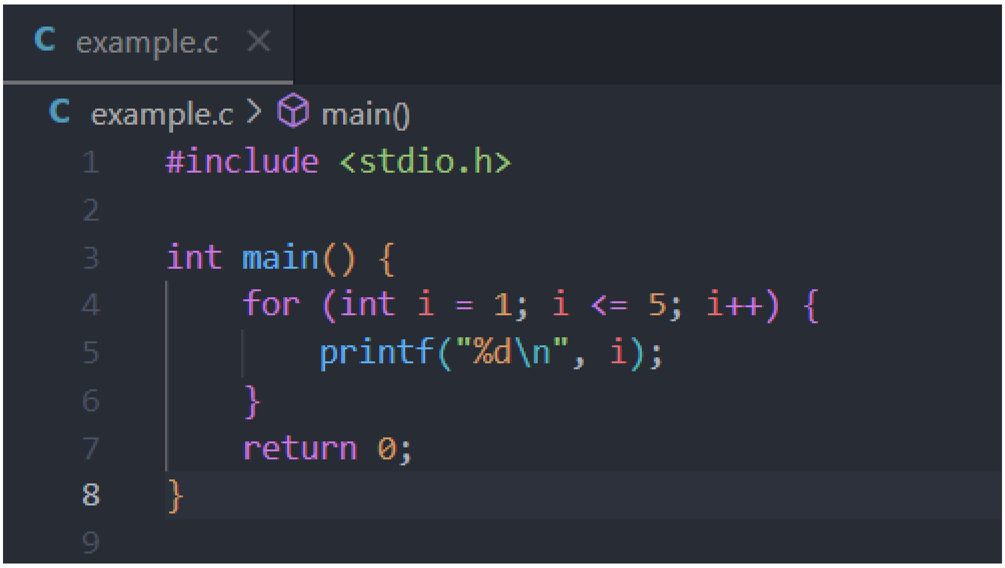
https://jmeubank.github.io/tdm-gcc/download/

Download the second option:

1. Open the project inside VScode
2. Install the following extensions:

- C/C++
- C/C++ Extension Pack

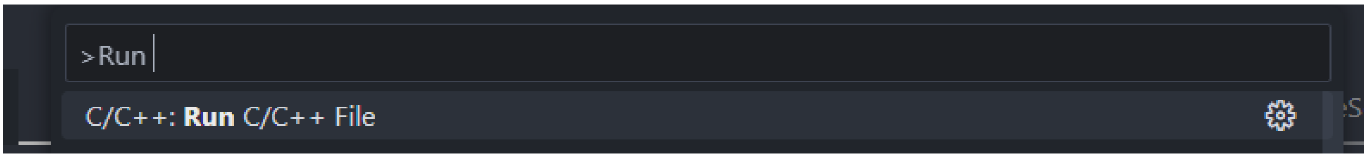
3. Open your c code file



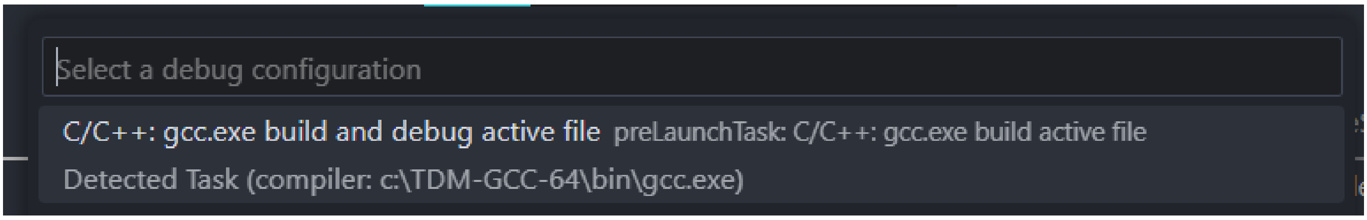
```c
#include <stdio.h>

int main() {
    for (int i = 1; i <= 5; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```

4. Press `Ctrl + Shift + P` to open the command palette
5. Type `Run C/C++ file` and select `Run C/C++ file`



6. Choose the gcc compiler



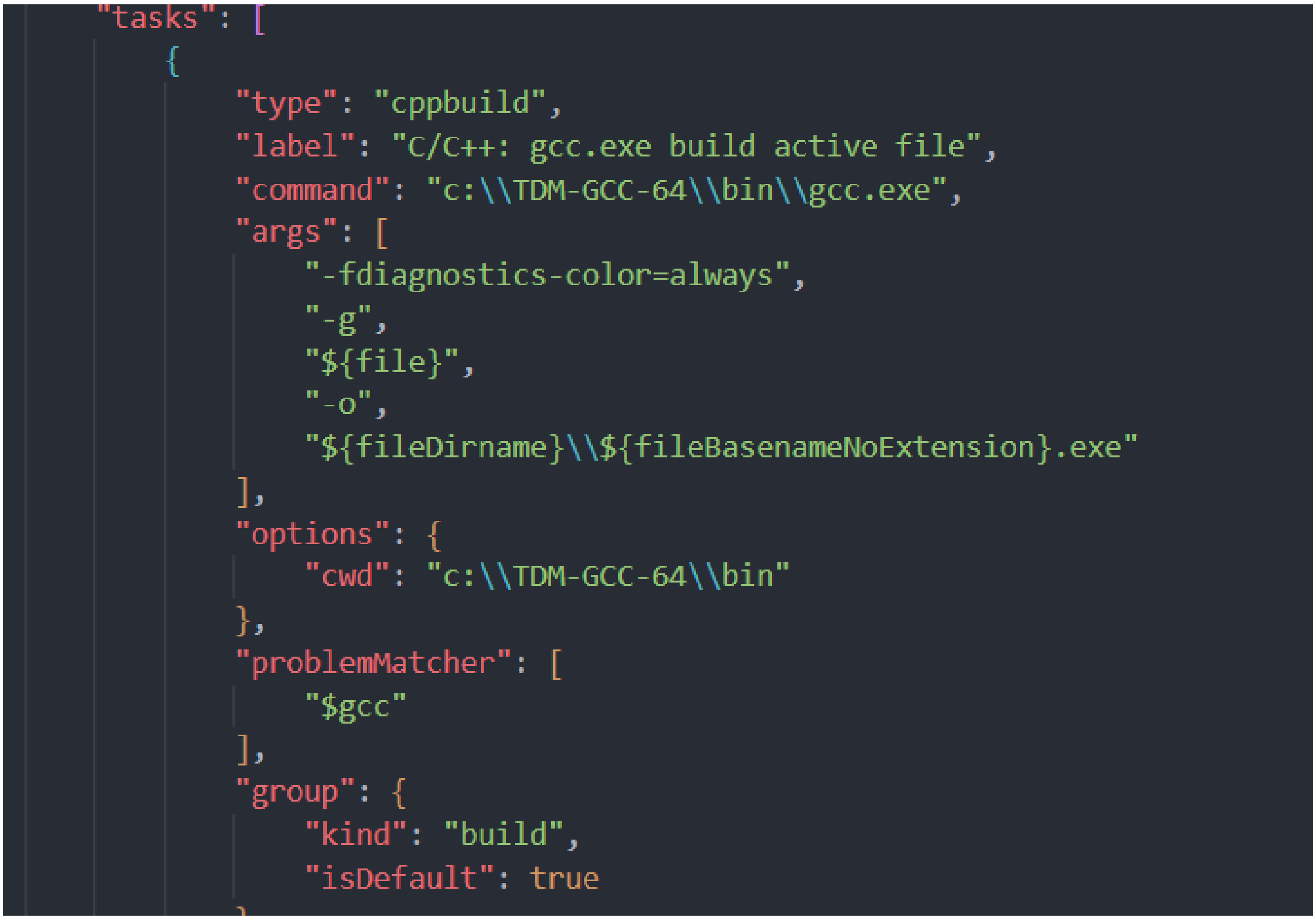
Select a debug configuration

C/C++: gcc.exe build and debug active file preLaunchTask: C/C++: gcc.exe build active file

Detected Task (compiler: c:\TDM-GCC-64\bin\gcc.exe)

7. A new folder named `.vscode` will be created in your project directory with the file `tasks.json`

8. Open the file `tasks.json`



```
    "tasks": [
        {
            "type": "cppbuild",
            "label": "C/C++: gcc.exe build active file",
            "command": "c:\\TDM-GCC-64\\bin\\gcc.exe",
            "args": [
                "-fdiagnostics-color=always",
                "-g",
                "${file}",
                "-o",
                "${fileDirname}\\${fileBasenameNoExtension}.exe"
            ],
            "options": {
                "cwd": "c:\\TDM-GCC-64\\bin"
            },
            "problemMatcher": [
                "$gcc"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
```

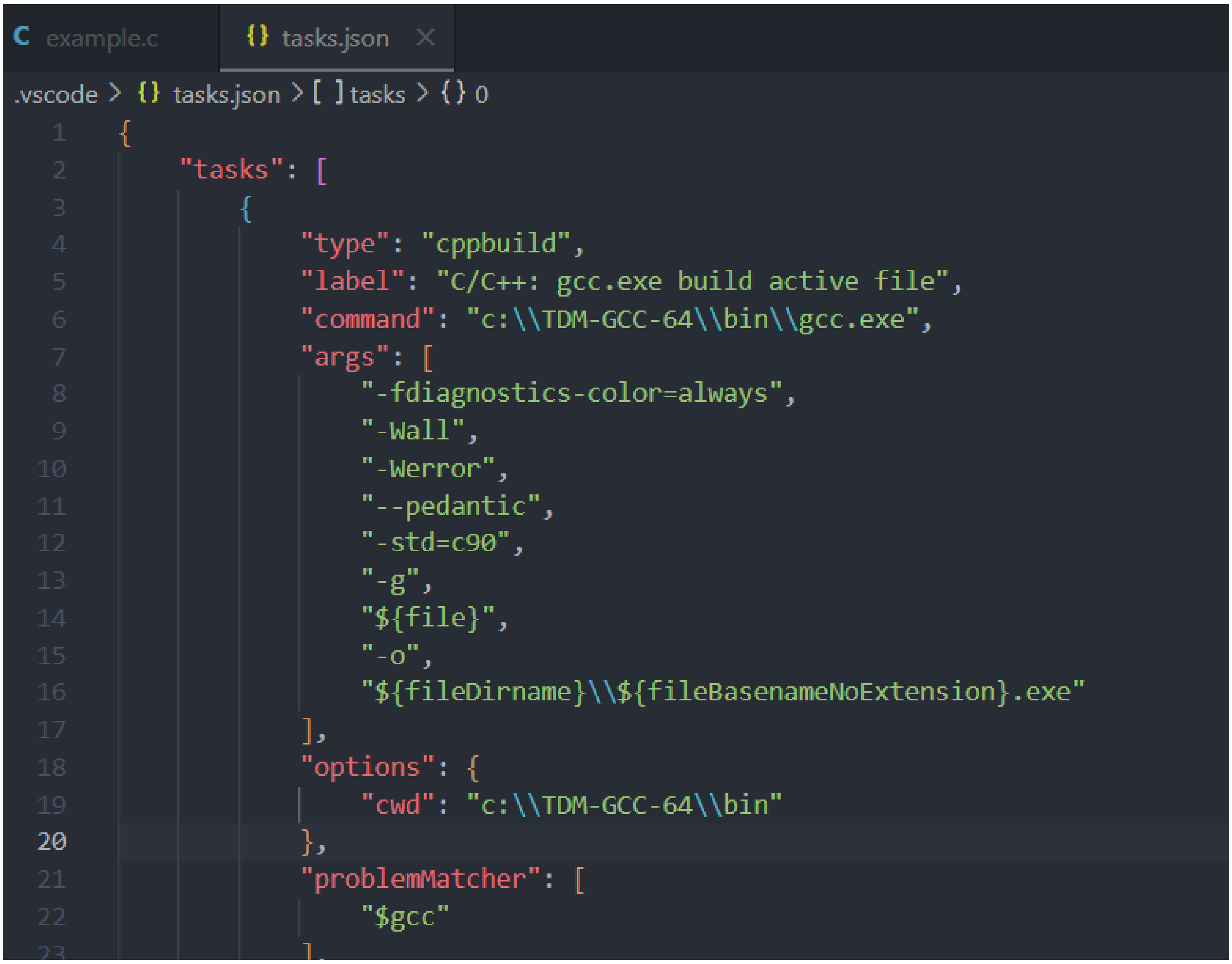
9. Add the following strings to the `args` array

~~~
"-Wall",
"-Werror",
"--pedantic",
"-std=c90",

The output will look like this:

```json
{
    "tasks": [
        {
            "type": "cppbuild",
            "label": "C/C++: gcc.exe build active file",
            "command": "c:\\TDM-GCC-64\\bin\\gcc.exe",
~~~

```
            "args": [
                "-fdiagnostics-color=always",
                "-Wall",
                "-Werror",
                "--pedantic",
                "-std=c90",
                "-g",
                "${file}",
                "-o",
                "${fileDirname}\\${fileBasenameNoExtension}.exe"
            ],
            "options": {
                "cwd": "c:\\TDM-GCC-64\\bin"
            },
            "problemMatcher": [
                "$gcc"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "detail": "Task generated by Debugger."
        }
    ],
    "version": "2.0.0"
}
```

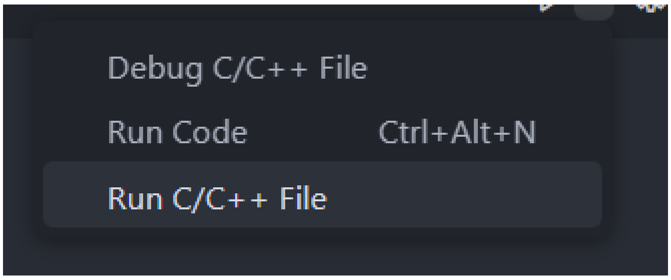


```
{
    "tasks": [
        {
            "type": "cppbuild",
            "label": "C/C++: gcc.exe build active file",
            "command": "c:\\TDM-GCC-64\\bin\\gcc.exe",
            "args": [
                "-fdiagnostics-color=always",
                "-Wall",
                "-Werror",
                "--pedantic",
                "-std=c90",
                "-g",
                "${file}",
                "-o",
                "${fileDirname}\\${fileBasenameNoExtension}.exe"
            ],
            "options": {
                "cwd": "c:\\TDM-GCC-64\\bin"
            },
            "problemMatcher": [
                "$gcc"
            ],
```

Dont forget to check, whether the `command` points to the correct GCC compiler

10. Press `Ctrl + S` to save the file

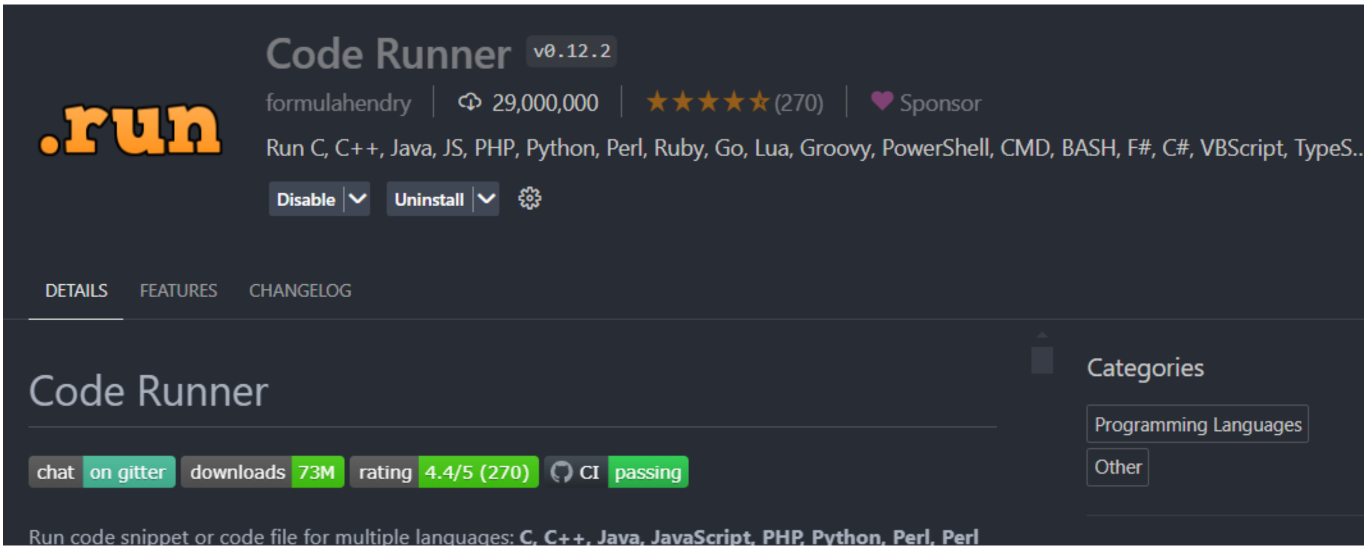
11. Run the code again by clicking on:



Alternative method:

Download the following extension:

- Code Runner



Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, CMD, BASH, F#, C#, VBScript, TypeS..

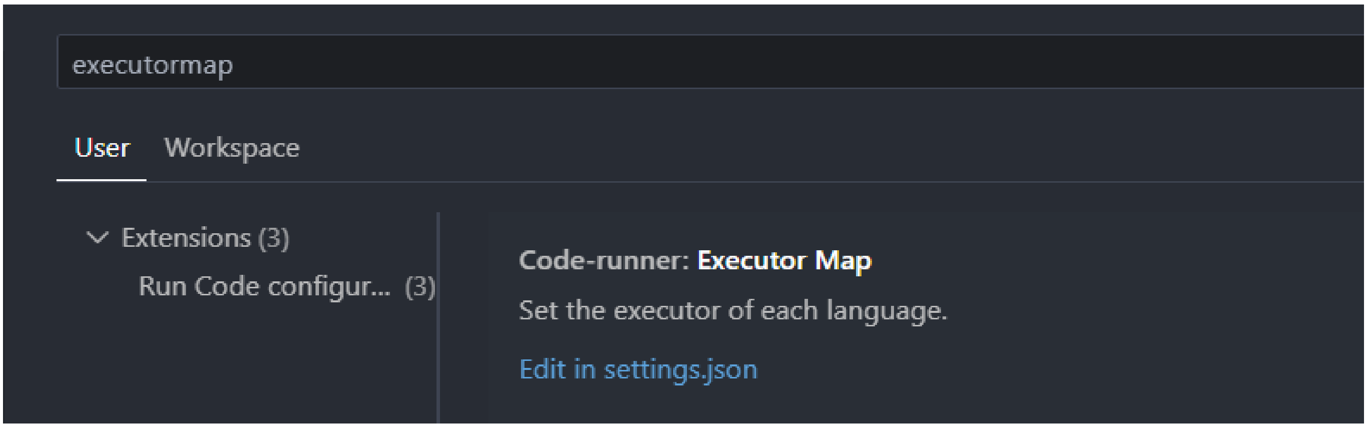
Code Runner

Run code snippet or code file for multiple languages: **C, C++, Java, JavaScript, PHP, Python, Perl, Perl**

Click on File -> Preferences -> Settings

Type in the search bar:

executormap



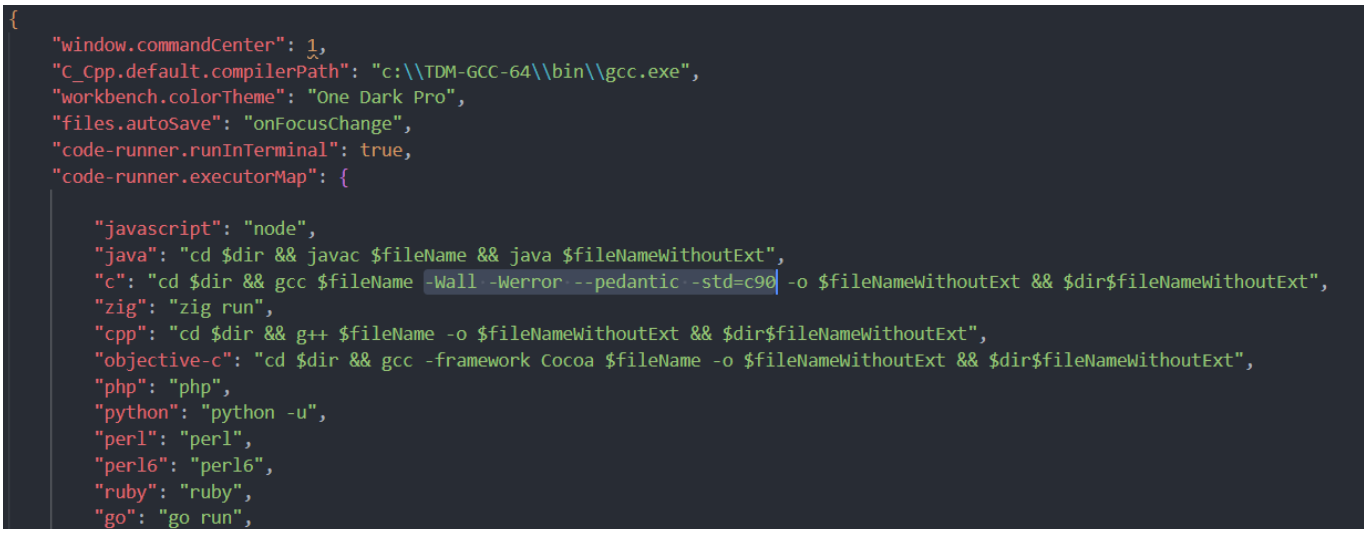
Code-runner: **Executor Map**

Set the executor of each language.

Edit in settings.json

Open the file and add:

-Wall -Werror --pedantic -std=c90

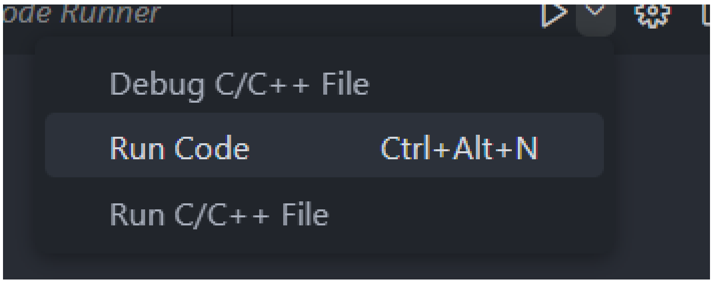


```
{
    "window.commandCenter": 1,
    "C_Cpp.default.compilerPath": "c:\\TDM-GCC-64\\bin\\gcc.exe",
    "workbench.colorTheme": "One Dark Pro",
    "files.autoSave": "onFocusChange",
    "code-runner.runInTerminal": true,
    "code-runner.executorMap": {

        "javascript": "node",
        "java": "cd $dir && javac $fileName && java $fileNameWithoutExt",
        "c": "cd $dir && gcc $fileName -Wall -Werror --pedantic -std=c90 -o $fileNameWithoutExt && $dir$fileNameWithoutExt",
        "zig": "zig run",
        "cpp": "cd $dir && g++ $fileName -o $fileNameWithoutExt && $dir$fileNameWithoutExt",
        "objective-c": "cd $dir && gcc -framework Cocoa $fileName -o $fileNameWithoutExt && $dir$fileNameWithoutExt",
        "php": "php",
        "python": "python -u",
        "perl": "perl",
        "perl6": "perl6",
        "ruby": "ruby",
        "go": "go run",
```

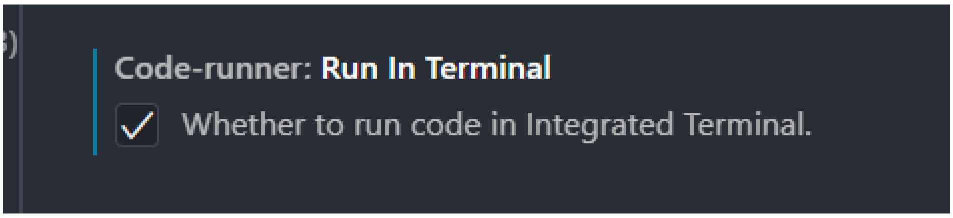
Save the file and go to your c code file

Click on Run code:



How to enable writing inside terminal for scanf etc.

Type in code runner inside the settings search bar and mark the following options:



Code-runner: **Run In Terminal**

☑ Whether to run code in Integrated Terminal.