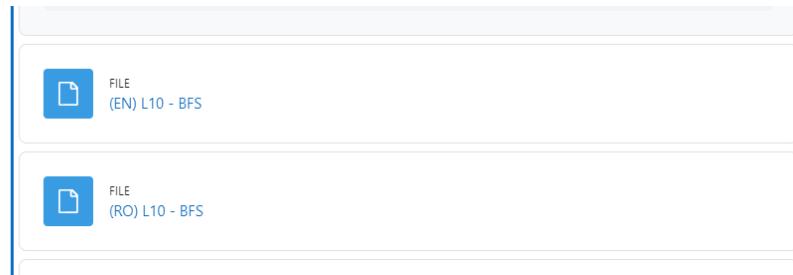


1 Assignment 9: Tutorial

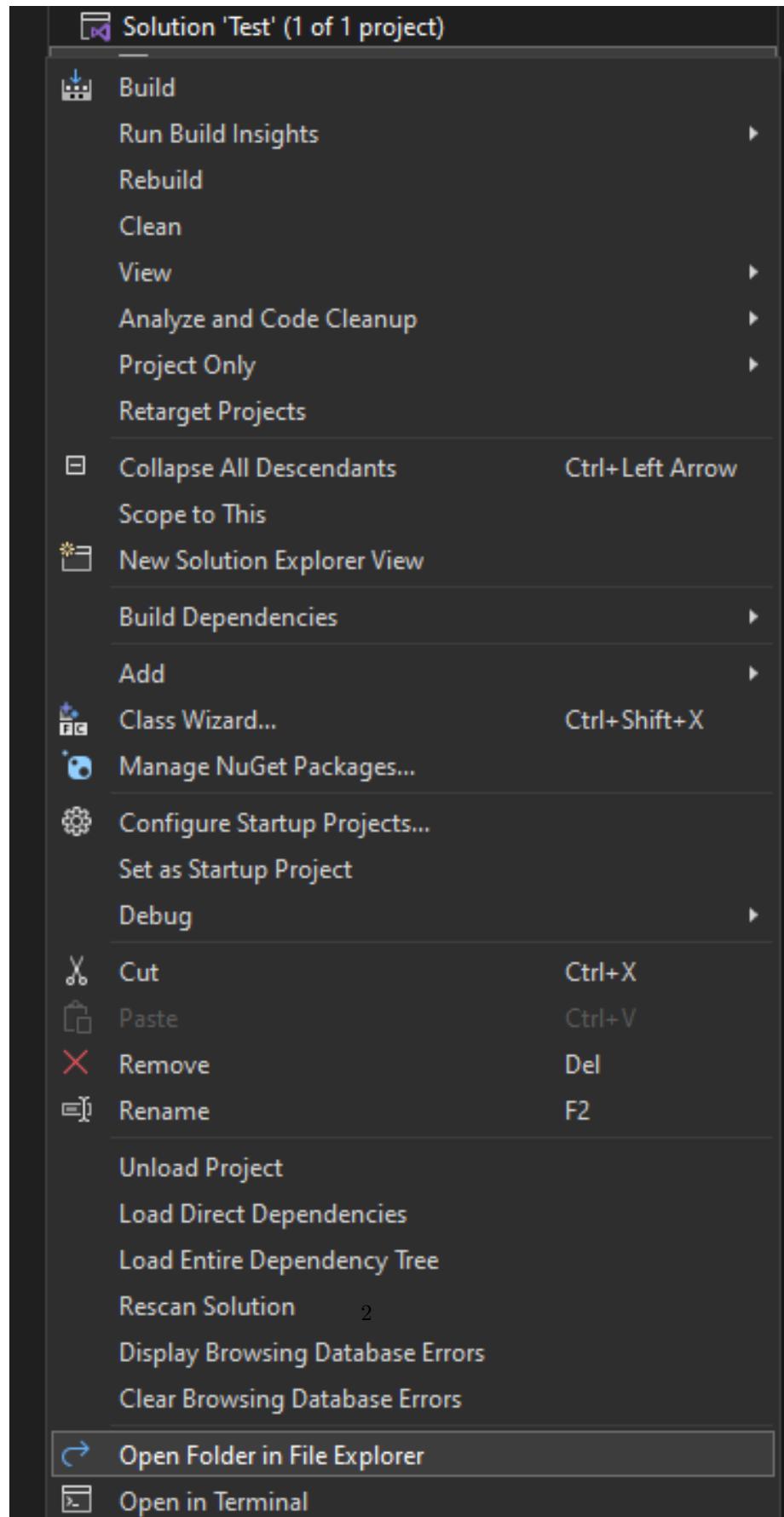
1.1 Visual Studio Project Setup

1. Go to moodle and select one of the following:

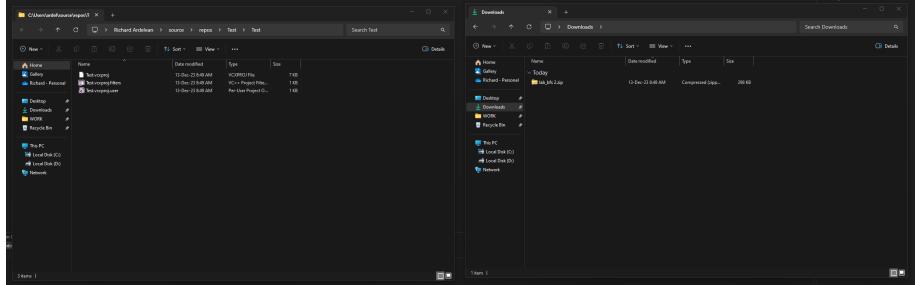


This will result in a '.zip' file being downloaded.

2. Create an Empty C++ Project and by right-clicking on the project (not the solution) you will be able to select 'Open Folder in File Explorer'.



- At this point, a ‘File Explorer’ window will be opened. Open another ‘File Explorer’ with the location of your downloads (most likely the Downloads folder).



- Open the ‘.zip’ file and copy as shown below the files from the zip file to the folder of the project.

The image displays two screenshots of Windows File Explorers. The top screenshot shows the contents of a zip file named 'lab_bfs_2.zip' located in the 'Downloads' folder. The bottom screenshot shows the 'Text' project folder in Visual Studio's Solution Explorer. The files copied from the zip file are highlighted in blue in both windows.

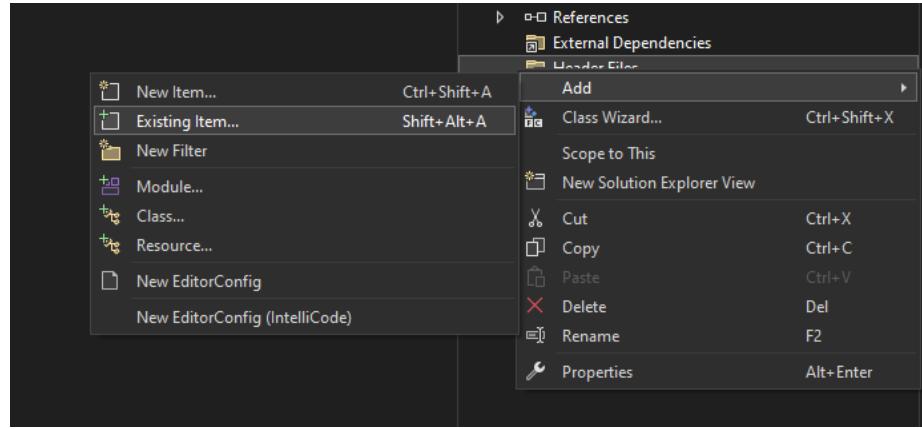
Screenshot 1: Content of lab_bfs_2.zip

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
DS_Store	DS_STORE File	1 KB	No	7 KB	98%	05-Dec-19 9:59 AM
bfs.cpp	CPP File	3 KB	No	8 KB	72%	29-Nov-19 4:47 PM
bfs.h	H File	1 KB	No	1 KB	56%	28-Nov-19 5:01 PM
grid.txt	Text Document	1 KB	No	1 KB	82%	18-Nov-19 6:58 PM
lab_bfs_eng.pdf	Chrome HTML Document	171 KB	No	187 KB	9%	05-Dec-19 9:59 AM
main.cpp	CPP File	3 KB	No	15 KB	80%	29-Nov-19 2:46 PM
Makefile	File	1 KB	No	1 KB	40%	23-Nov-19 11:02 AM
Profiler.h	H File	117 KB	No	1,068 KB	90%	07-Oct-19 6:04 PM

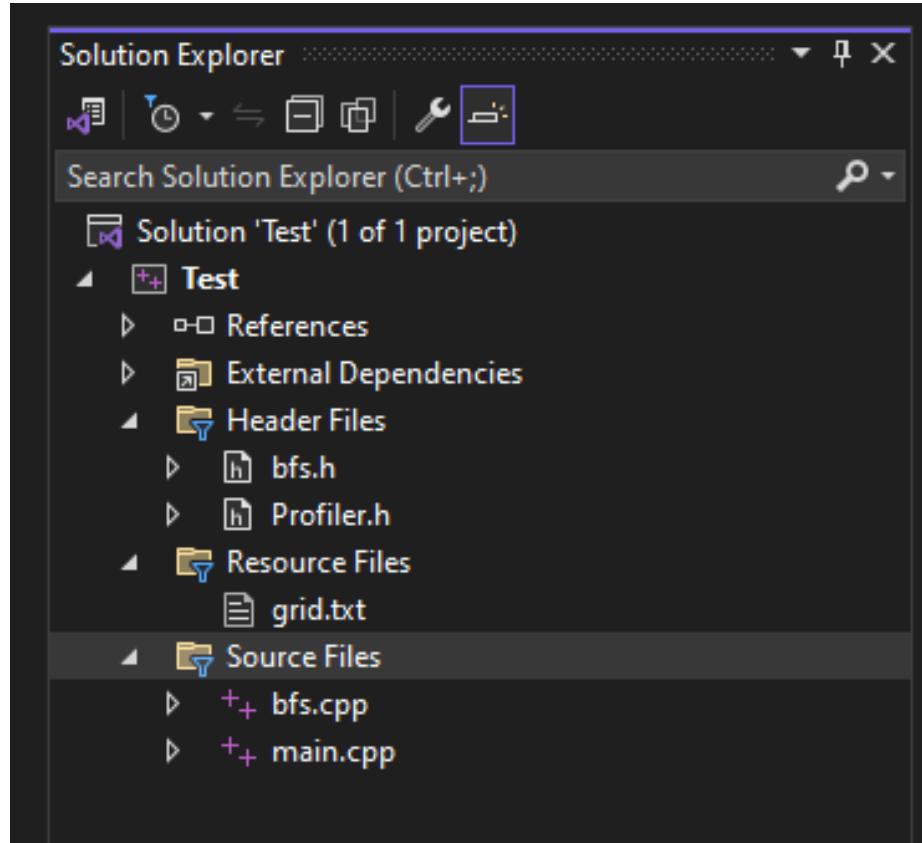
Screenshot 2: Visual Studio Solution Explorer

Name	Date modified	Type	Size
grid.h	15-Dec-19 8:49 AM	VCProject File	7 KB
grid.txt	15-Dec-19 8:49 AM	Text Document	1 KB
grid.h	15-Dec-19 8:49 AM	File for Project DS	1 KB
grid.h	15-Dec-19 8:49 AM	File	208KB
grid.h	15-Dec-19 8:49 AM	File	1 KB
grid.h	15-Dec-19 8:49 AM	Text Document	1 KB
grid.h	15-Dec-19 8:49 AM	CPP File	1 KB

- By selecting the folders from the Visual Studio Solution Explorer ‘Header Files / Resources / Source Files’ use the following options:

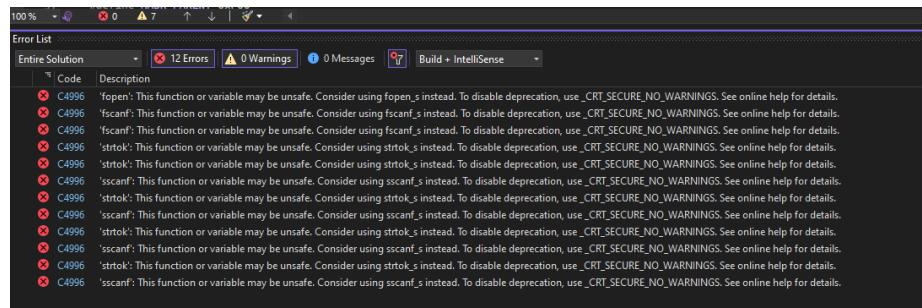


6. Such that, the following setup is accomplished:



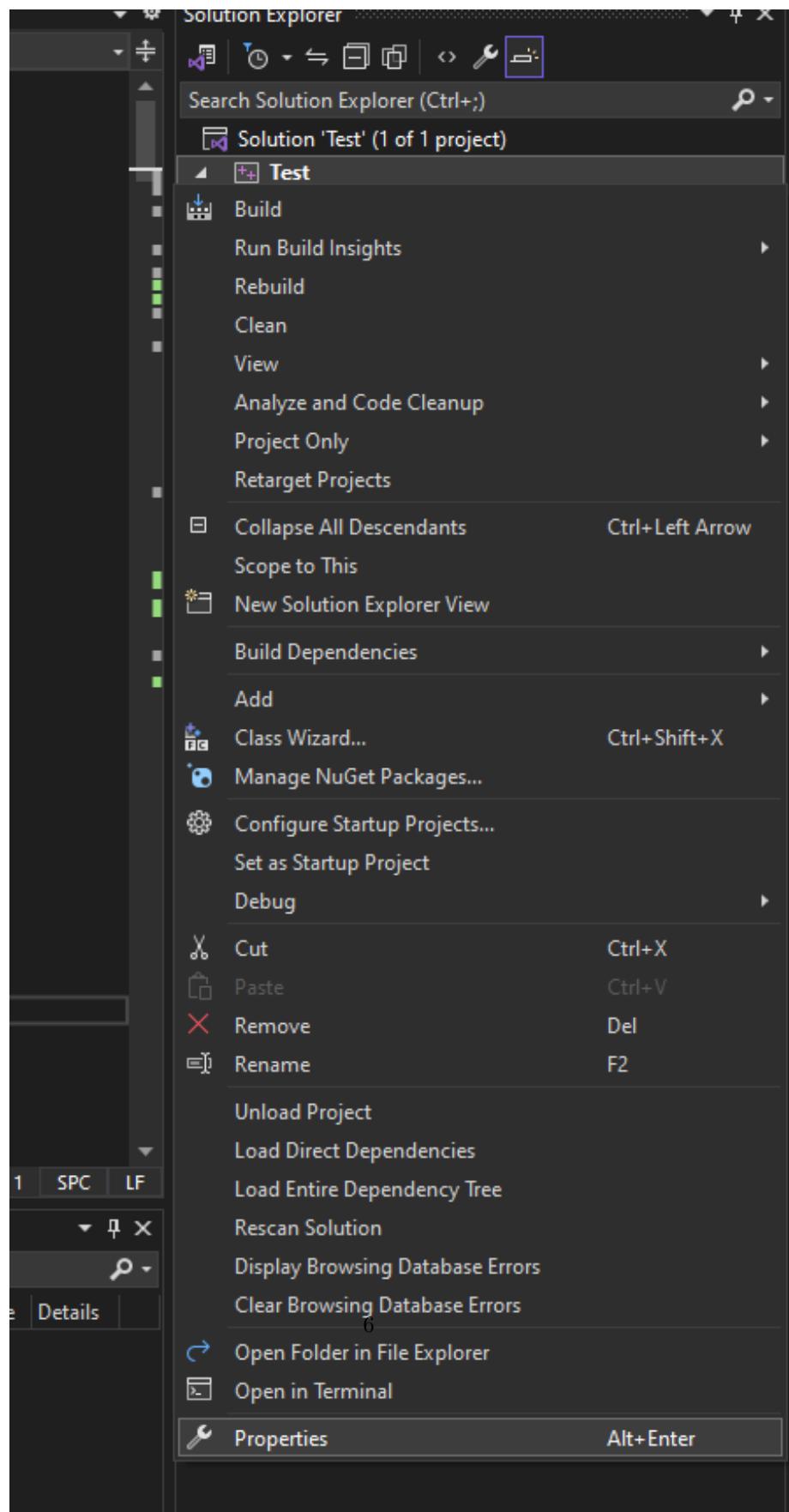
1.2 Visual Studio ‘unsafe’ error

Example:



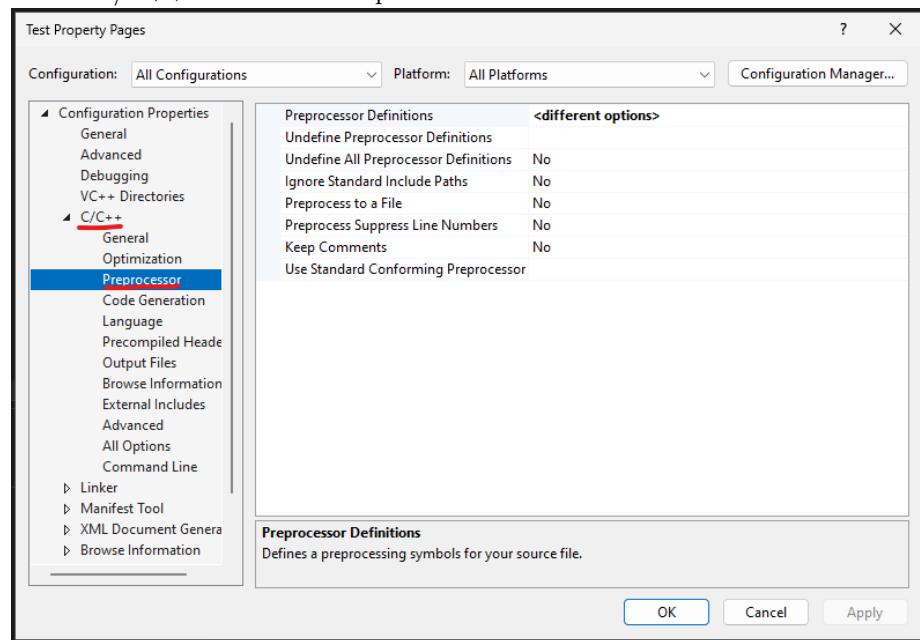
Solution:

Right-click on the project (not the Solution which will most likely have the same name) and select 'Properties'.

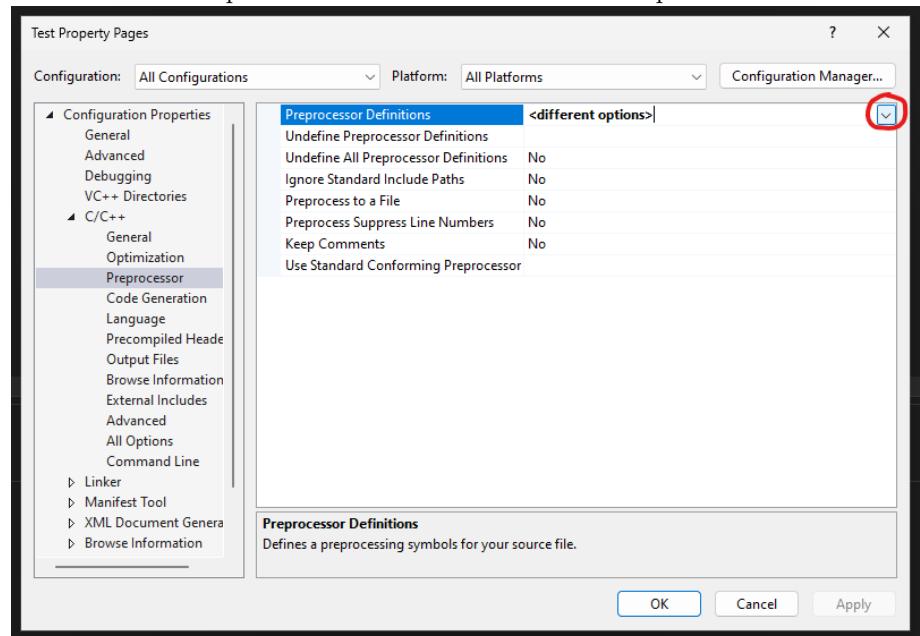


Update the ‘Configuration’ and ‘Platform’ in the upper side of the window to ‘All Configurations’ and ‘All Platforms’, respectively.

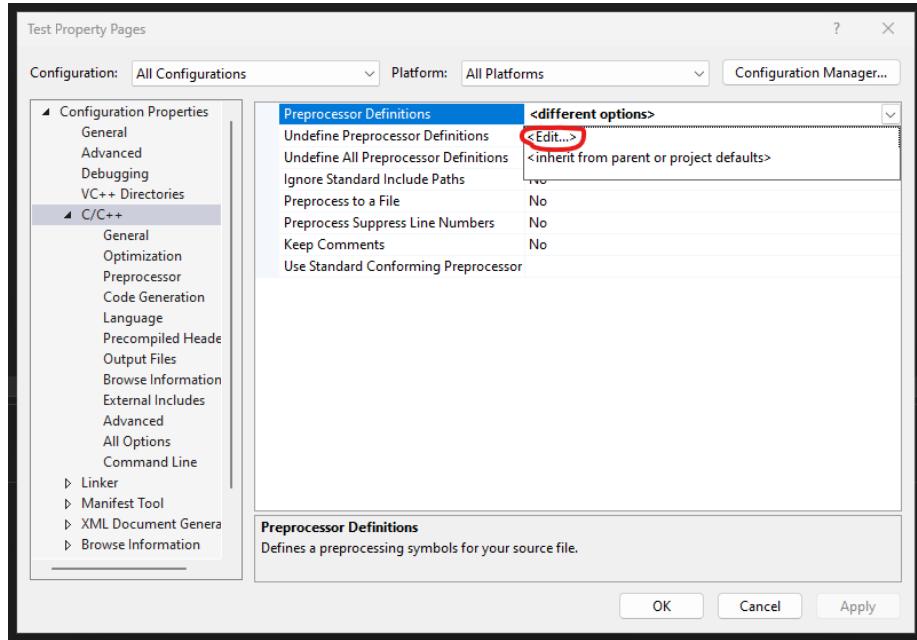
Go to ‘C/C++’ and select ‘Preprocessor’.



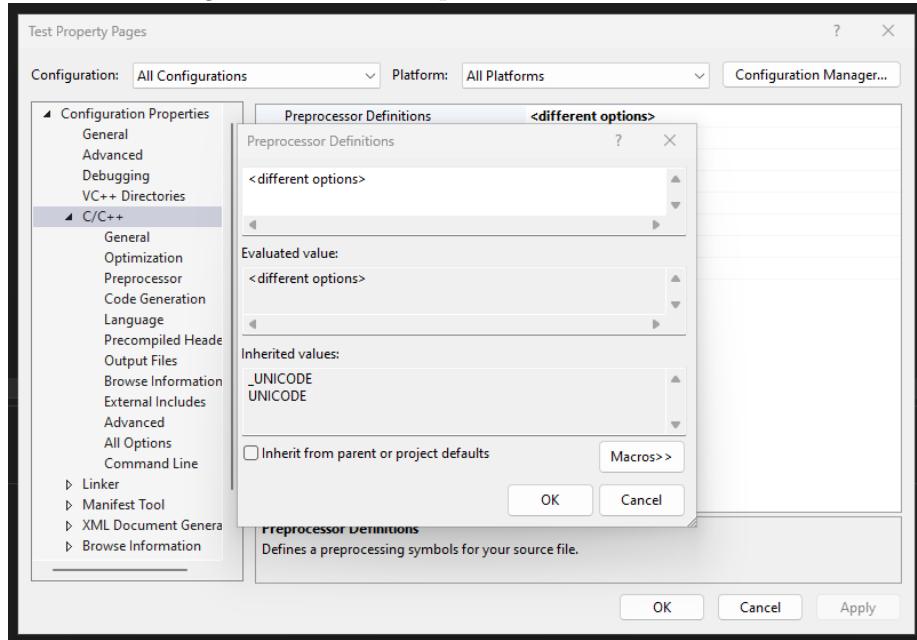
Then click on ‘Preprocessor Definitions’ and on the dropdown arrow.



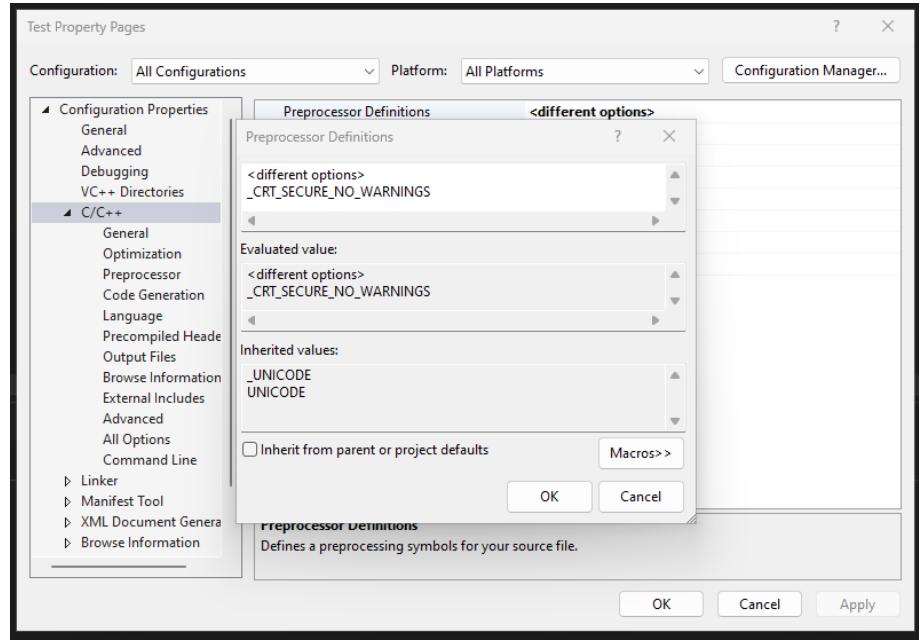
Select ‘Edit’



And the following new window will open:



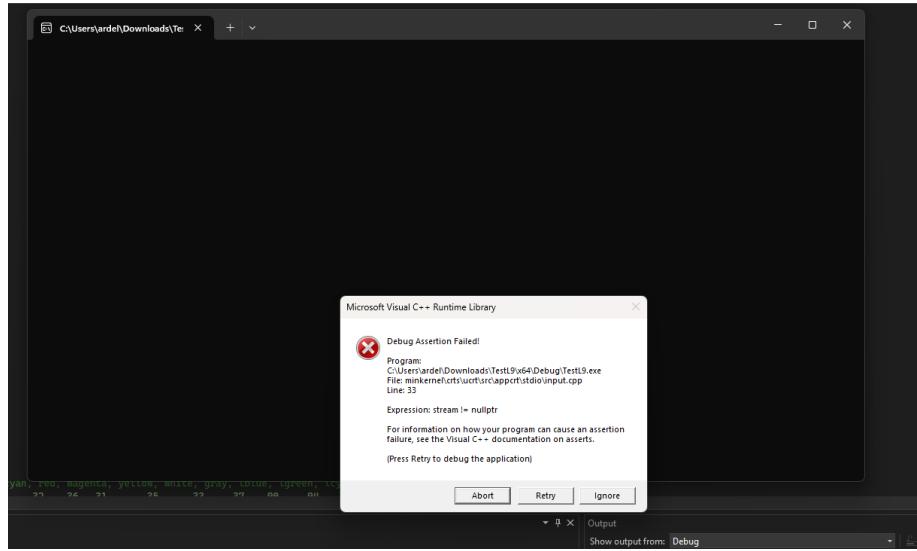
Introduce '_CRT_SECURE_NO_WARNINGS' under '<different options>' as shown below.



Click on ‘OK’ for all opened windows until you are returned to the main Visual Studio window of the project, and you will be able to run the project.

1.3 Visual Studio ‘Assertion’ Error

This indicates that you have not followed the tutorial. Go back to the first page and make sure that you have moved the files from the ‘Downloads’ folder to the ‘Project’ folder. You might also need to *remove* all the files from the Visual Studio IDE and *re-add* them by hand.

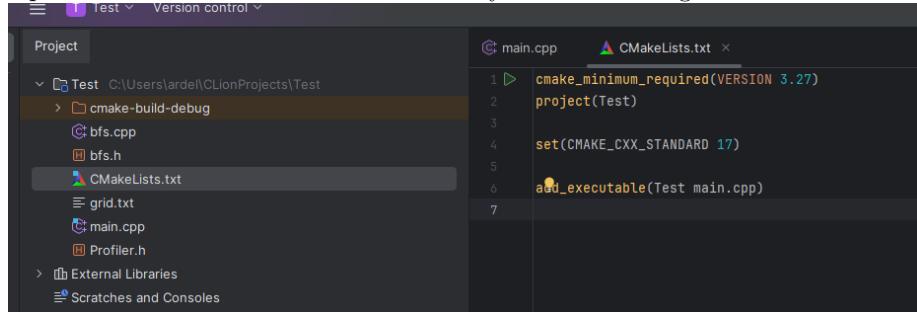


1.4 CLion undefined Error

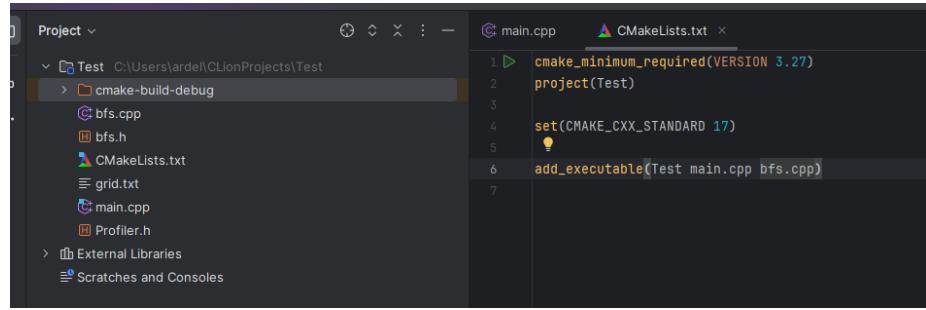
Example:

Solution:

Open the ‘CMakeLists.txt’ file and modify in the following manner:



```
add_executable(Test main.cpp)  
into  
add_executable(Test main.cpp bfs.cpp)
```

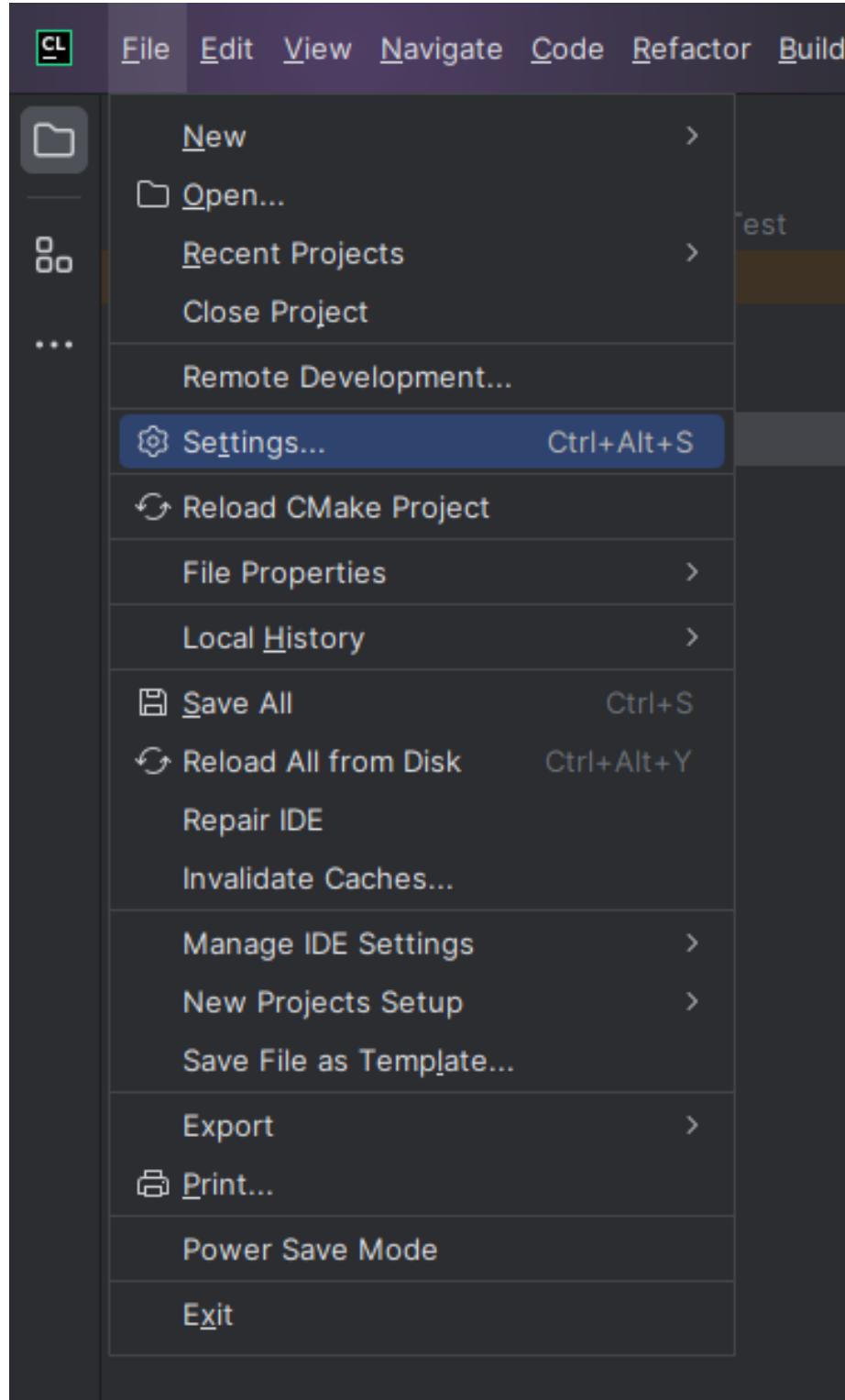


The screenshot shows the CLion IDE interface. On the left, the Project tool window displays a CMake project named 'Test' located at 'C:\Users\ardel\CLionProjects\Test'. The project structure includes a 'cmake-build-debug' folder containing files like 'bfs.cpp', 'bfs.h', 'CMakeLists.txt', 'grid.txt', 'main.cpp', and 'Profiler.h'. Below this are sections for 'External Libraries' and 'Scratches and Consoles'. On the right, the main editor window shows the 'CMakeLists.txt' file with the following content:

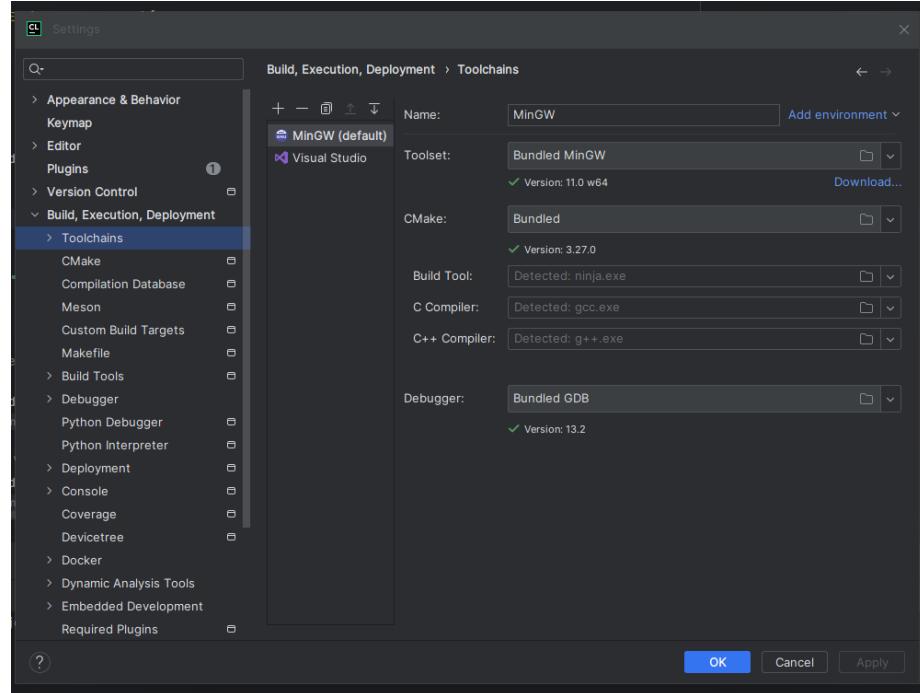
```
1 cmake_minimum_required(VERSION 3.27)
2 project(Test)
3
4 set(CMAKE_CXX_STANDARD 17)
5
6 add_executable(Test main.cpp bfs.cpp)
```

1.5 CLion Visual Studio – Option 1 (slower)

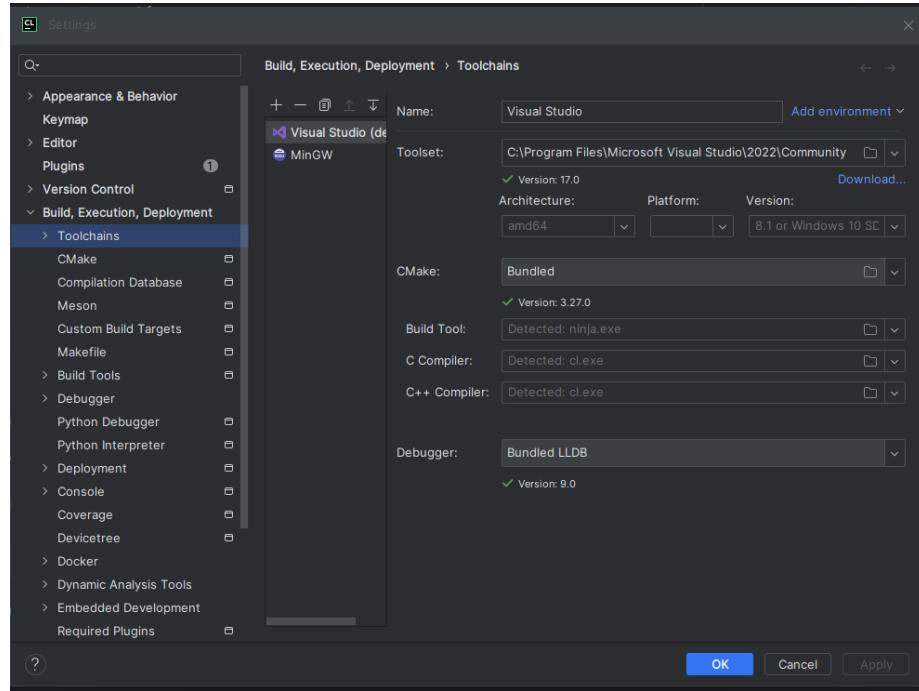
Go to File -> Settings:



In the newly opened window, go to ‘Build, Execution, Deployment’ -> ‘Toolchains’



Using the arrows, move Visual Studio as default:



1.6 CLion MinGW – Option 2 (faster, requires external console)

1.6.1 CLion Clear error

Example:

A screenshot of the CLion 'Run' tab. The tab bar shows 'Run' and 'Test'. The terminal window displays the command 'clear' being run in a directory path: 'C:\Users\ardel\CLionProjects\Test\cmake-build-debug'. The output shows an error message: "'clear' is not recognized as an internal or external command, operable program or batch file." Navigation icons for the terminal are visible on the left.

Solution:

Go to the main.cpp file and scroll to lines 113-117 in the displayGrid function.

The screenshot shows the CLion IDE interface. On the left is the project tree, with 'Test' selected. Inside 'Test' are 'cmake-build-debug', 'bfs.cpp', 'bfs.h', 'CMakeLists.txt', 'grid.txt', 'main.cpp', and 'Profiler.h'. Under 'External Libraries' and 'Scratches and Consoles' are empty entries. On the right is the code editor for 'main.cpp'. The code is as follows:

```

98     return "/\\";
99 }else if((x & MASK_PARENT) == MASK_DOWN){
100    return "\\";
101 }else if((x & MASK_PARENT) == MASK_LEFT){
102    return "<";
103 }else if((x & MASK_PARENT) == MASK_RIGHT){
104    return ">";
105 }else{
106    return " ";
107 }
108 }
109 void displayGrid(const Grid *grid, int lastCommand)
110 {
111    int i, j;
112 #ifdef _MSC_VER
113    system("cls");
114 #else
115    system( Command: "clear");
116 #endif
117 }

```

Modify in the following manner:

In the else branch from line 116
`system("clear");`
 to
`system("cls");`

The screenshot shows the CLion IDE interface with the same project structure as the first screenshot. The code editor for 'main.cpp' now has the modification applied at line 116:

```

98     return "/\\";
99 }else if((x & MASK_PARENT) == MASK_DOWN){
100    return "\\";
101 }else if((x & MASK_PARENT) == MASK_LEFT){
102    return "<";
103 }else if((x & MASK_PARENT) == MASK_RIGHT){
104    return ">";
105 }else{
106    return " ";
107 }
108 }
109 void displayGrid(const Grid *grid, int lastCommand)
110 {
111    int i, j;
112 #ifdef _MSC_VER
113    system("cls");
114 #else
115    system( Command: "cls");
116 #endif
117 }

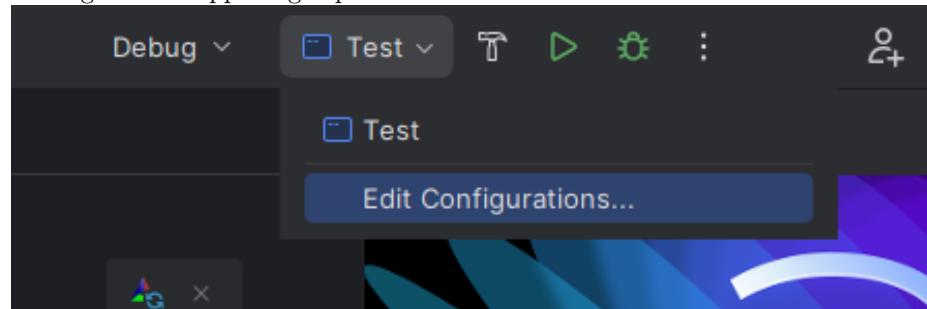
```

1.6.2 CLion not showing grid



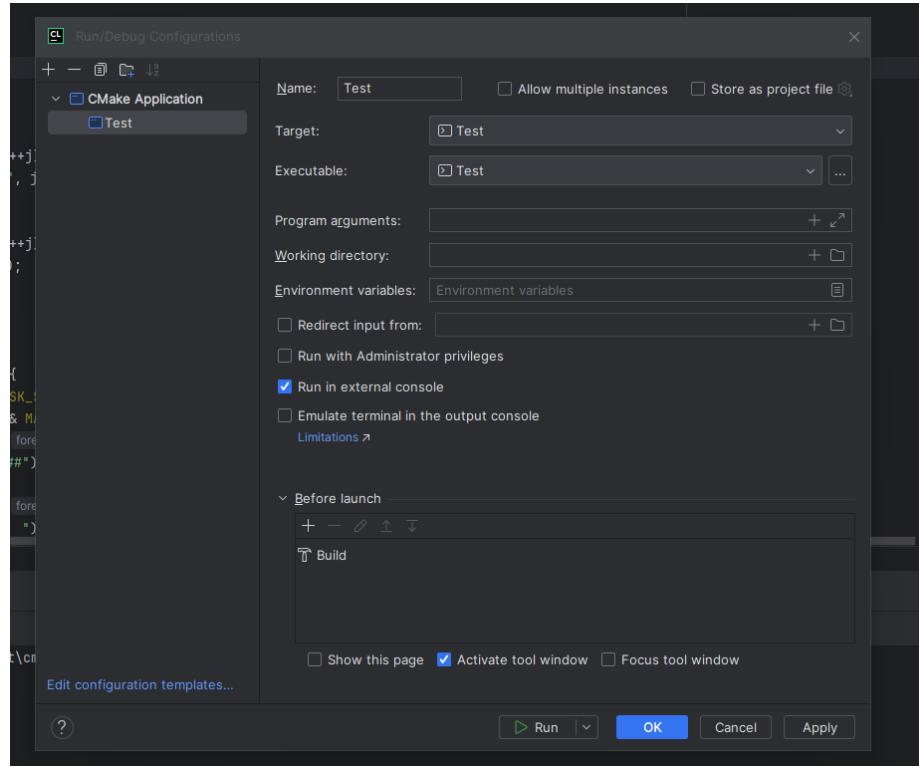
Copy the 'grid.txt' file into the 'cmake-build-debug' folder.

Then go to the upper right part of the screen:



Select 'Edit Configurations' and check the following boxes:

- Run in external console



1.7 Mac run command

```
g++ main.cpp bfs.cpp -std=c++11 && ./a.out
```