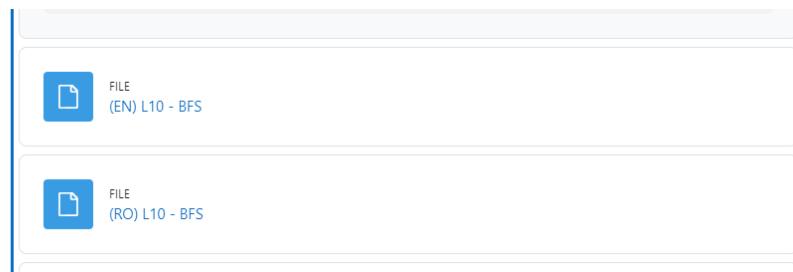


1 Tema Nr. 9: Tutorial

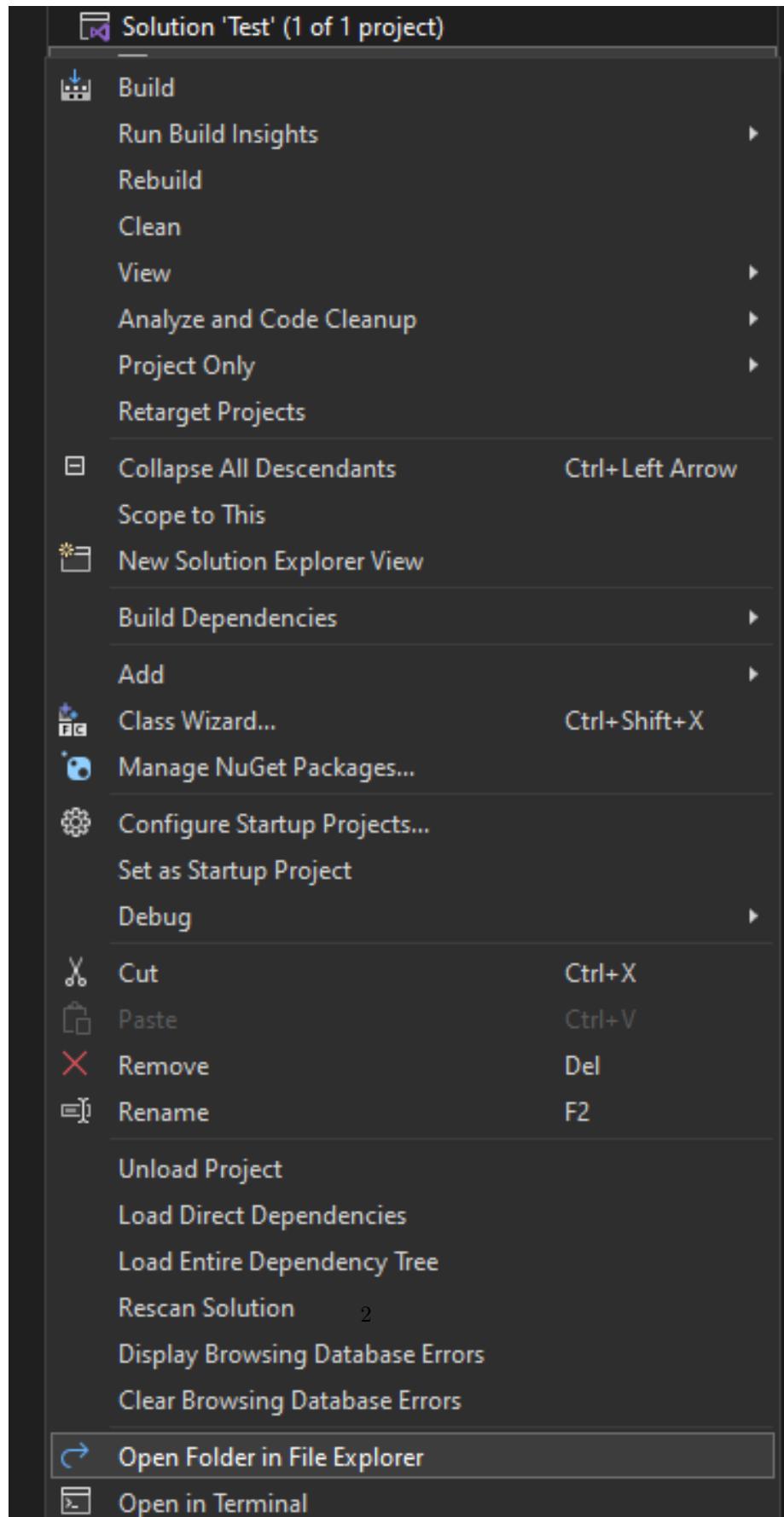
1.1 Configurare Proiect Visual Studio

1. Accesați Moodle și selectați una dintre următoarele:

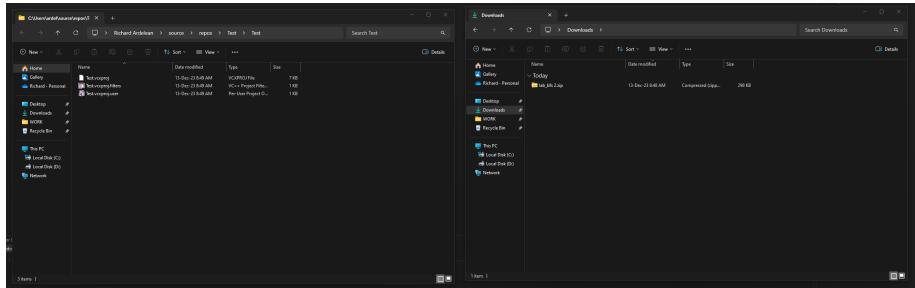


Aceasta va rezulta în descărcarea unui fișier '.zip'.

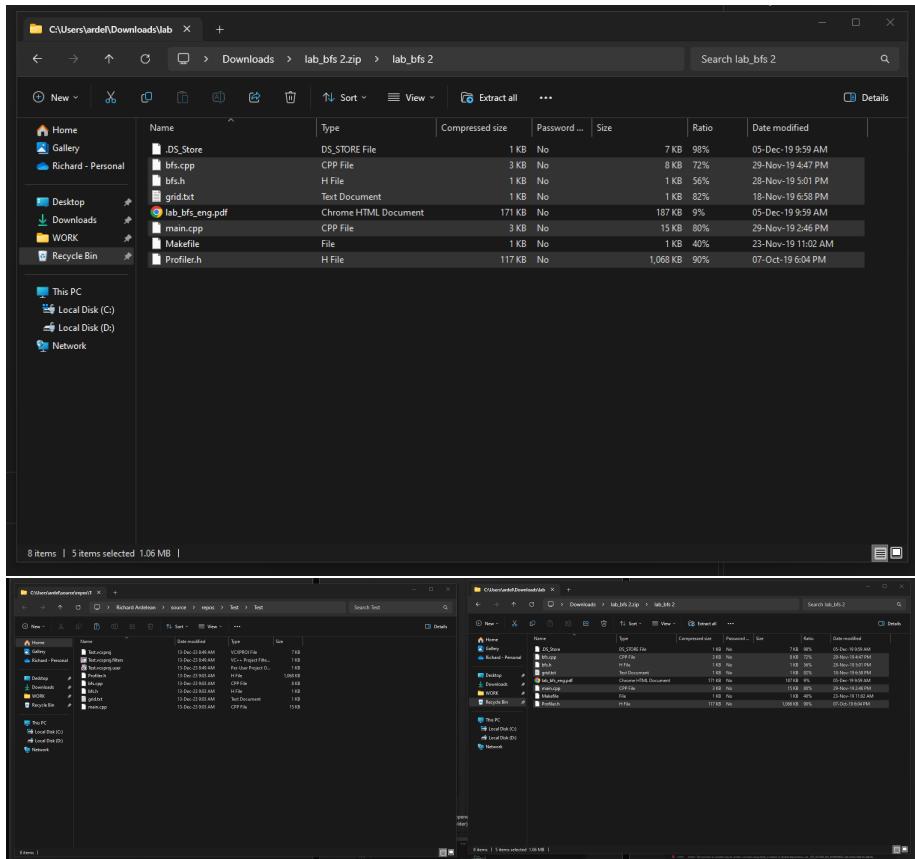
2. Creați un Proiect C++ Gol și făcând clic dreapta pe proiect (nu pe soluție) veți putea selecta 'Deschideți folderul în File Explorer'.



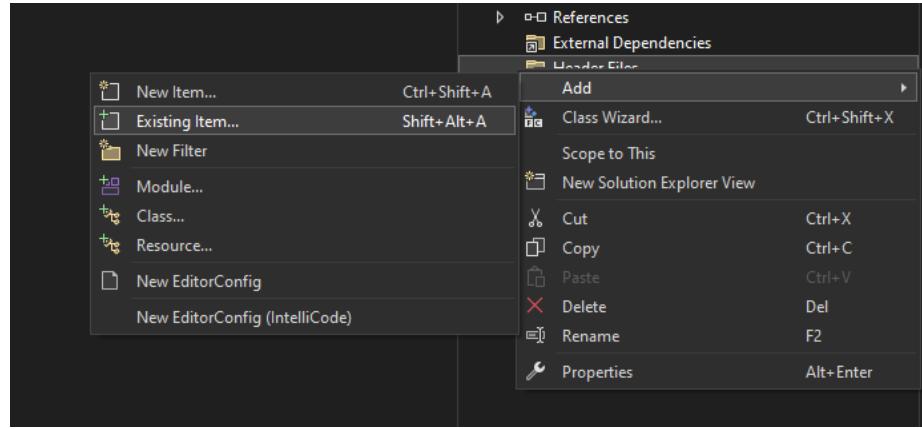
3. În acest moment, se va deschide o fereastră ‘File Explorer’. Deschideți o altă fereastră ‘File Explorer’ la locația descăr cărilor dvs. (cel mai probabil folderul Downloads).



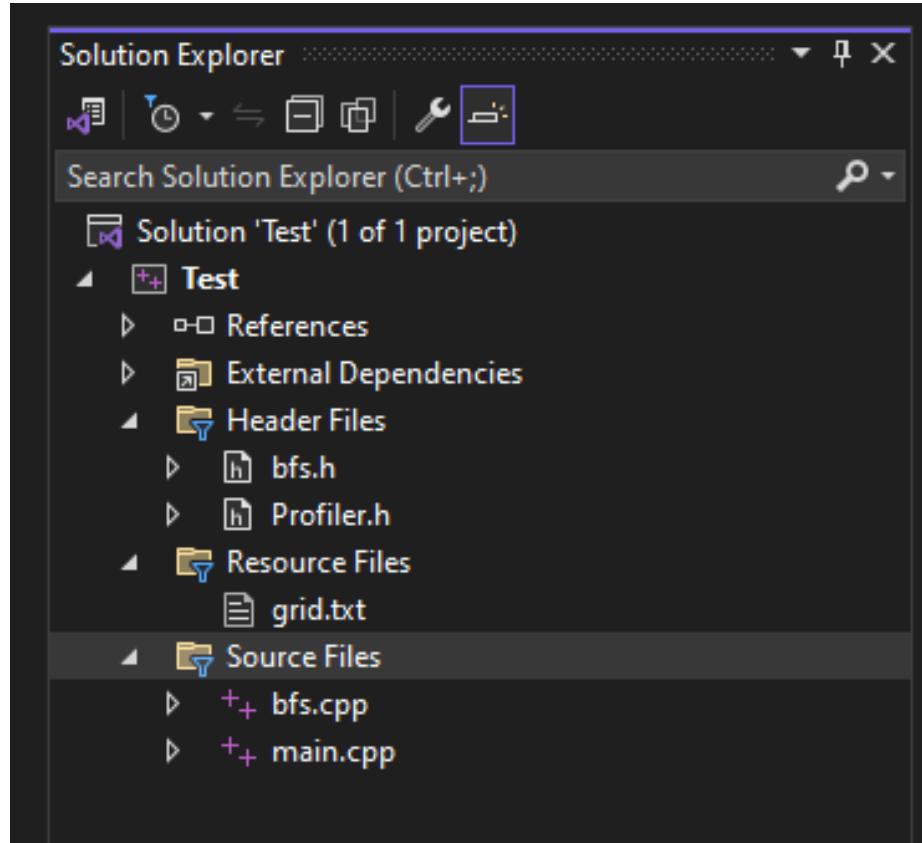
4. Deschideți fisierul ‘.zip’ și copiați, aşa cum se arată mai jos, fișierele din arhivă în folderul proiectului.



5. Selectând folderele din Solution Explorer al Visual Studio ‘Fișiere Header / Resurse / Fișiere Sursă‘ utilizați următoarele opțiuni:

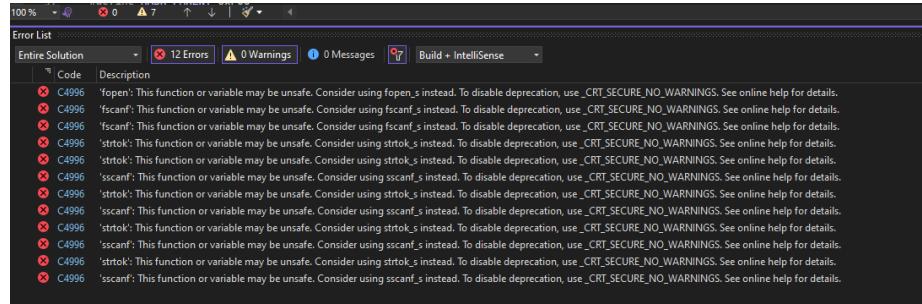


6. Astfel, se realizează următoarea structură:



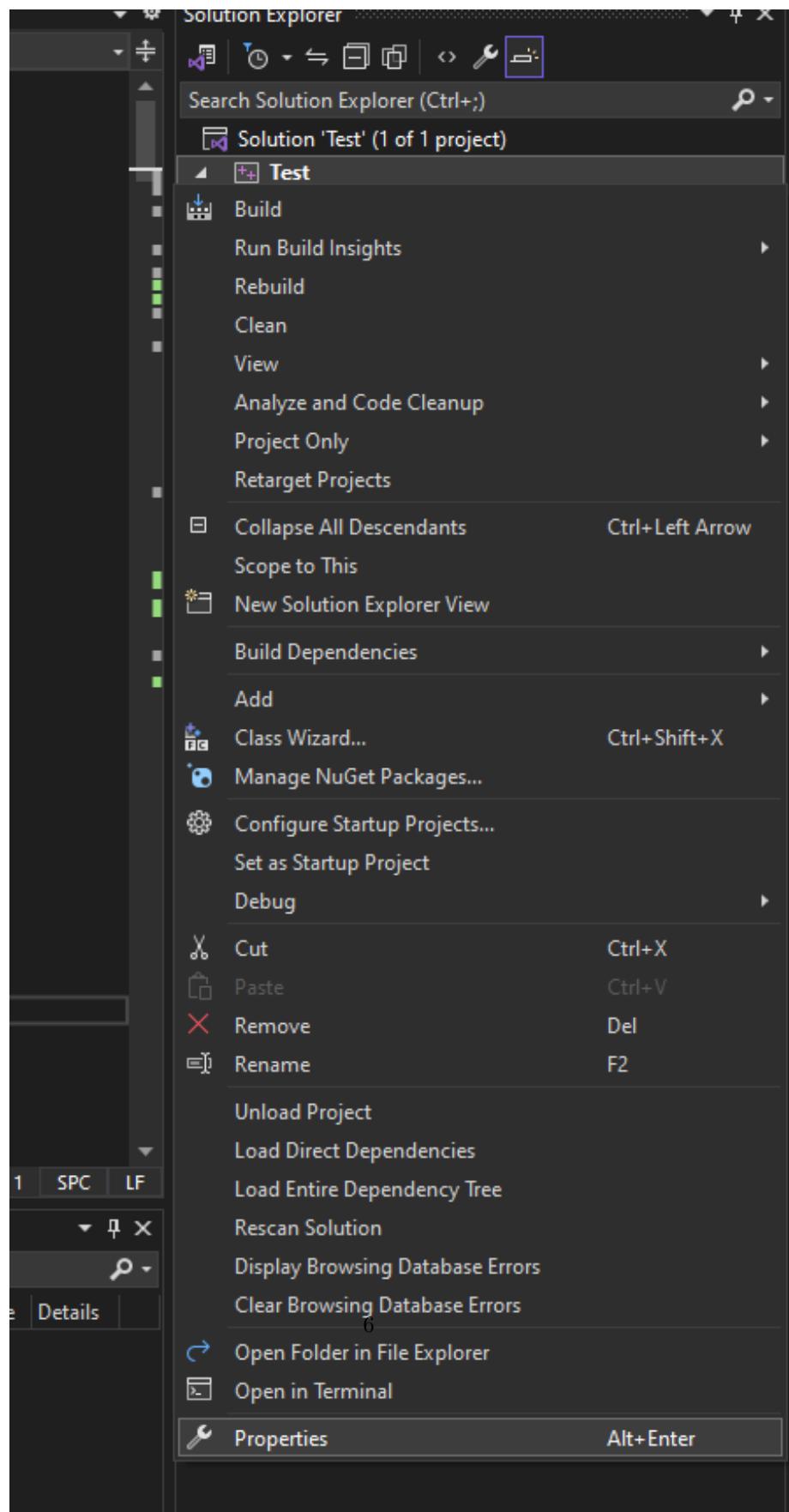
1.2 Eroare ‘unsafe’ în Visual Studio

Exemplu:



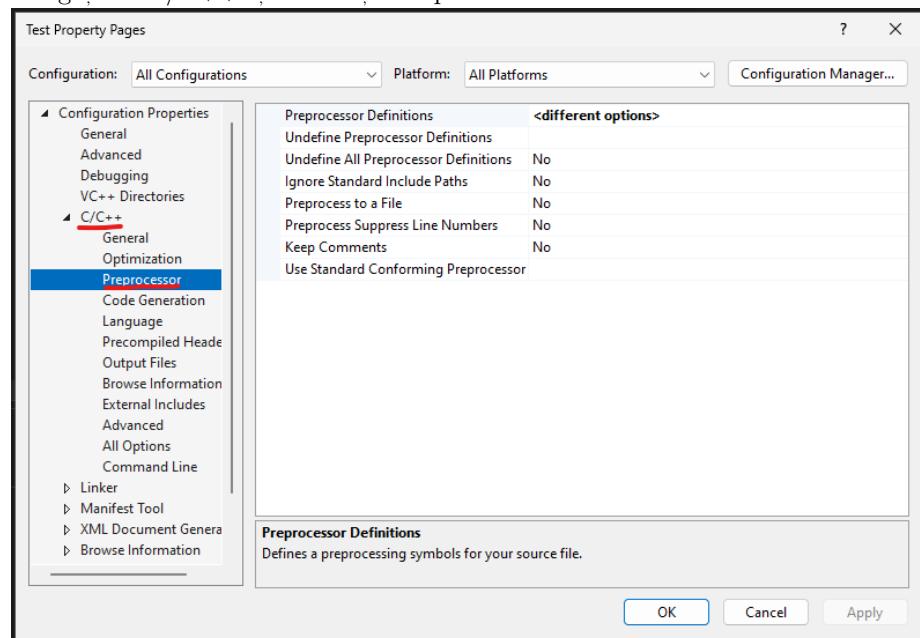
Soluție:

Faceți clic dreapta pe proiect (nu pe Soluție, care cel mai probabil are același nume) și selectați ‘Proprietăți’.

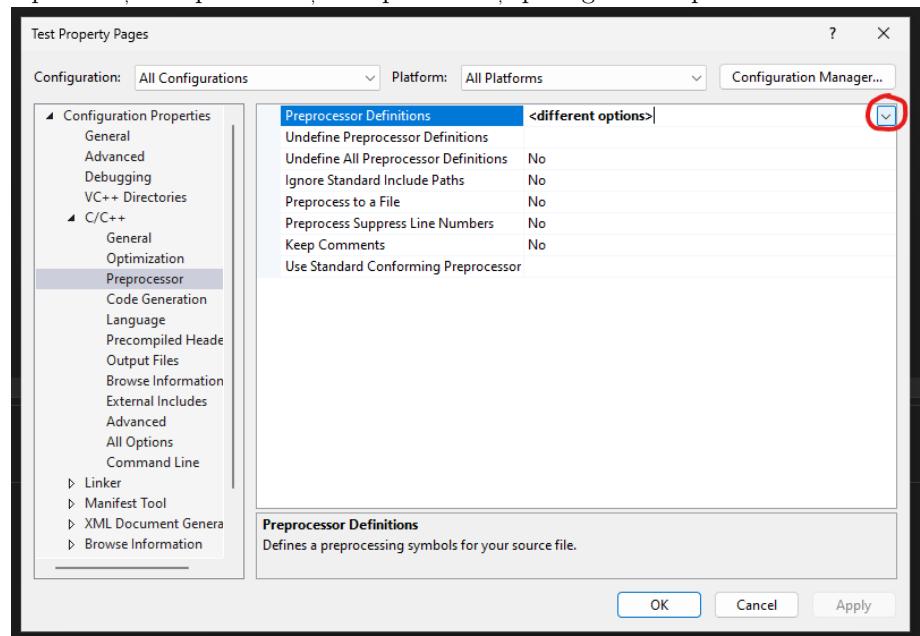


Actualizați ‘Configurație’ și ‘Platformă’ în partea superioară a ferestrei la ‘Toate configurațiile’ și ‘Toate platformele’, respectiv.

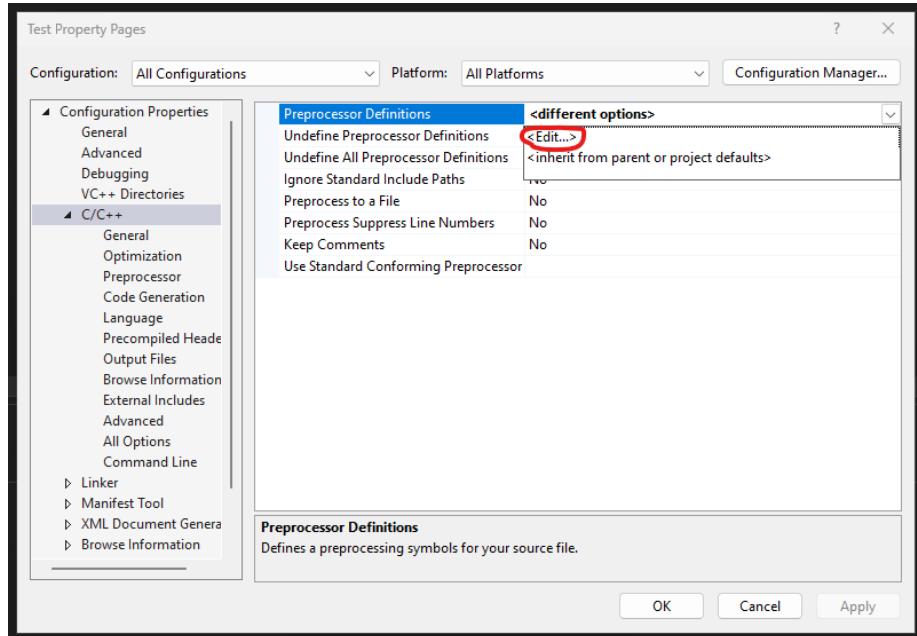
Mergeți la ‘C/C++’ și selectați ‘Preprocesor’.



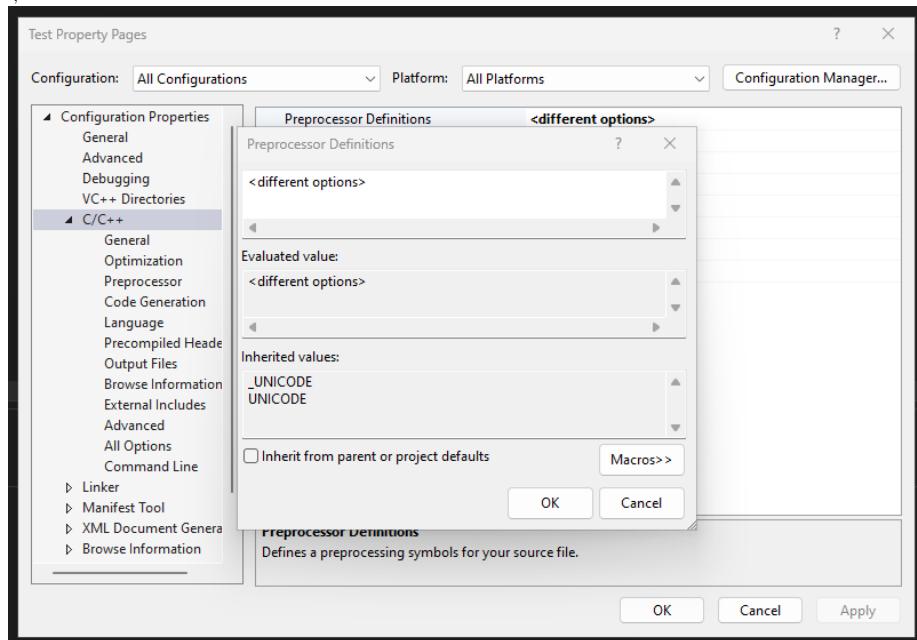
Apoi faceți clic pe ‘Definiții Preprocesor’ și pe săgeata dropdown.



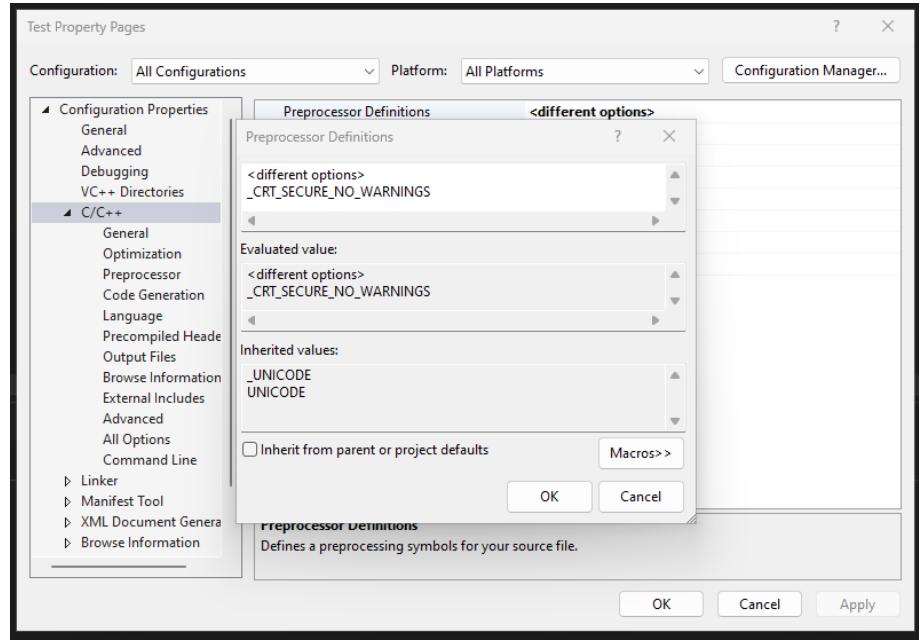
Selectați ‘Editare’



Si se va deschide următoarea fereastră nouă:



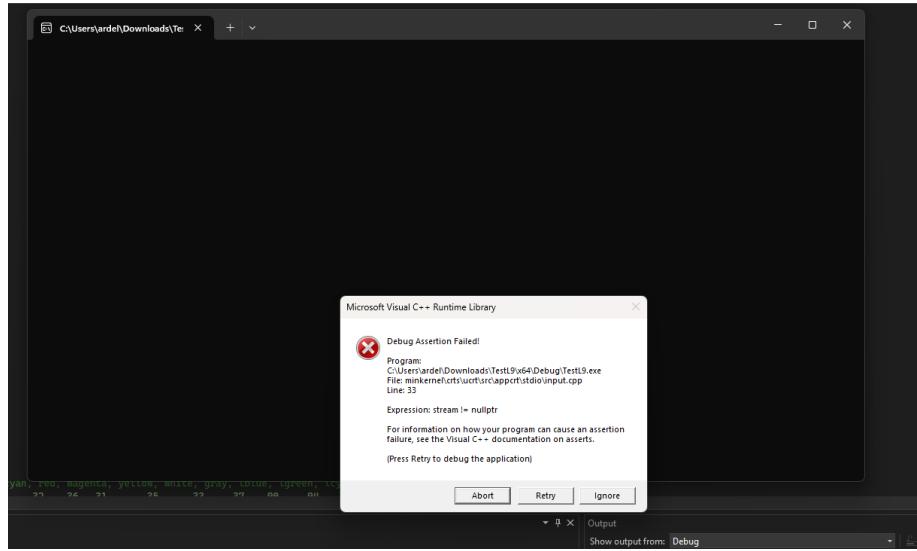
Introduceți ‘_CRT_SECURE_NO_WARNINGS’ sub ‘<opțiuni diferite>’, așa cum se arată mai jos.



Faceți clic pe ‘OK’ la toate ferestrele deschise până când reveniți la fereastra principală Visual Studio a proiectului și veți putea rula proiectul.

1.3 Eroare ‘Assertion’ în Visual Studio

Aceasta indică faptul că nu ați urmat tutorialul. Întoarceți-vă la prima pagină și asigurați-vă că ați mutat fișierele din folderul ‘Downloads’ în folderul ‘Project’. De asemenea, este posibil să fie nevoie să *ștergeți* toate fișierele din IDE-ul Visual Studio și să le *adăugati din nou* manual.



1.4 Eroare undefined în CLion

Exemplu:

Solutie:

Deschideți fișierul ‘CMakeLists.txt’ și modificați după cum urmează:

The screenshot shows the Clion IDE interface. The left sidebar displays the project structure under 'Test' (C:\Users\larde\CLionProjects\Test). It includes a 'cmake-build-debug' folder, source files like bfs.cpp, bfs.h, main.cpp, grid.txt, and Profiler.h, and configuration files CMakeLists.txt and External Libraries. The right panel shows the code editor with the file CMakeLists.txt open. The code defines a CMake project named 'Test' with version 3.27, sets the C++ standard to 17, and adds an executable named 'Test' from the main.cpp file.

```
cmake_minimum_required(VERSION 3.27)
project(Test)

set(CMAKE_CXX_STANDARD 17)

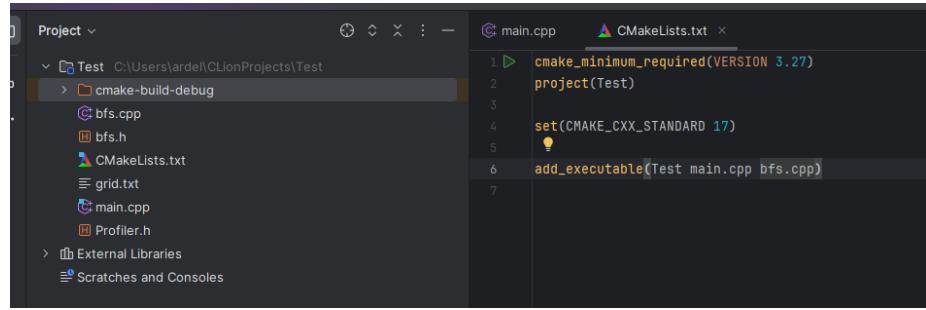
add_executable(Test main.cpp)
```

adăugați

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

in

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

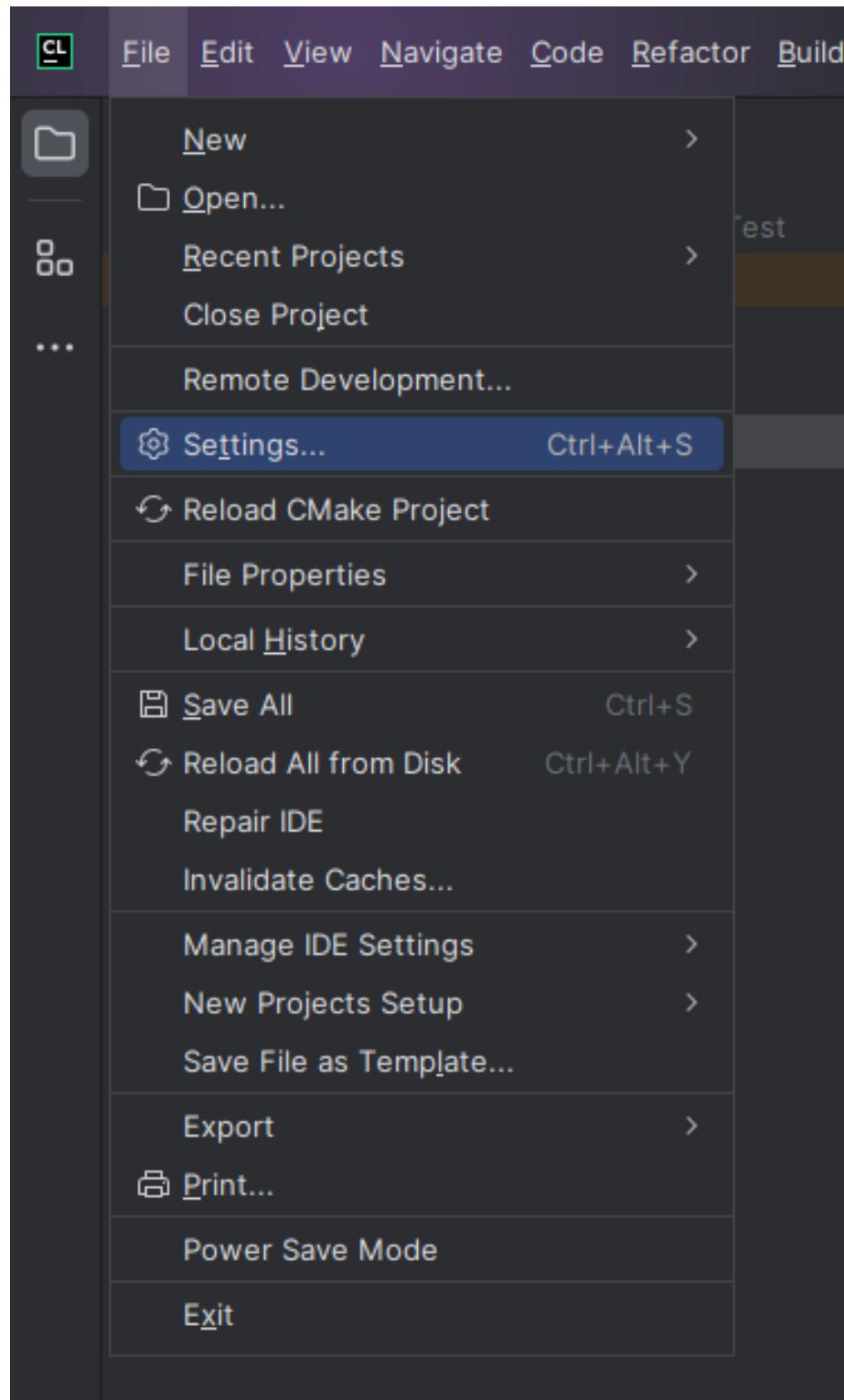


The screenshot shows the CLion IDE interface. On the left, the Project tool window displays a CMake project named 'Test' located at 'C:\Users\ardel\CLionProjects\Test'. Inside the project, there is a 'cmake-build-debug' build configuration folder containing files: 'bfs.cpp', 'bfs.h', 'CMakeLists.txt', '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:

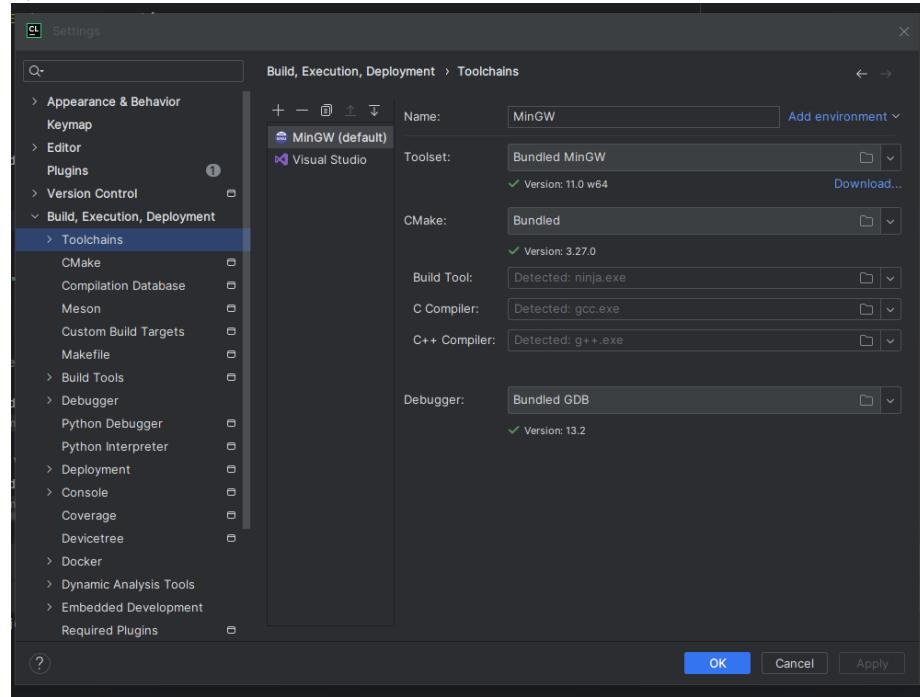
```
1 cmake_minimum_required(VERSION 3.27)
2 project(Test)
3
4 set(CMAKE_CXX_STANDARD 17)
5
6 add_executable(Test main.cpp bfs.cpp)
```

1.5 CLion Visual Studio – Opțiunea 1 (mai lentă)

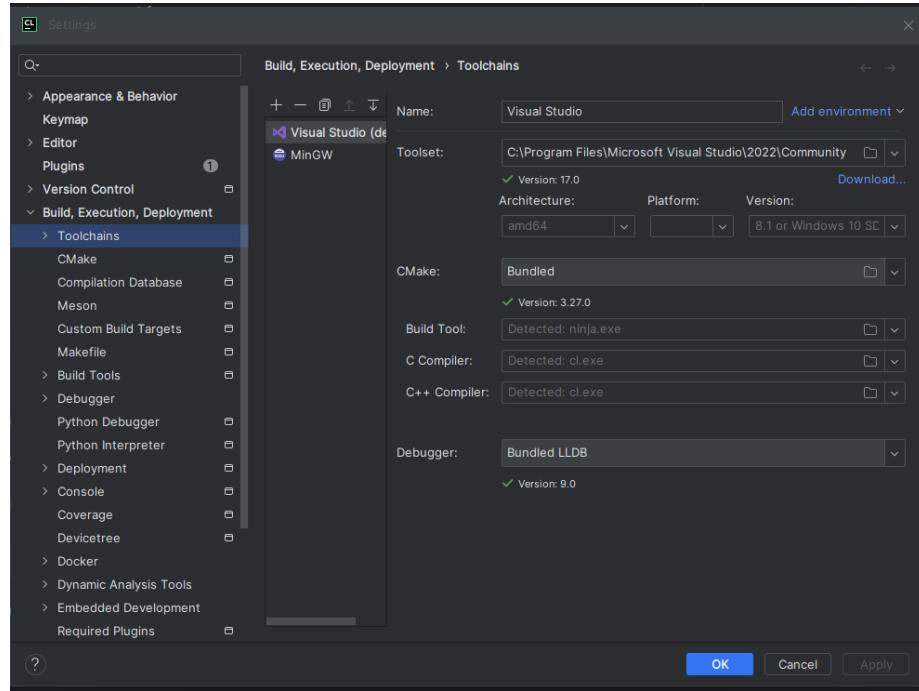
Mergeți la ‘Fisier’ -> ‘Setări’:



În fereastra nou deschisă, mergeți la ‘Compilare, Execuție, Implementare’ -> ‘Lanțuri de instrumente’



Folosind săgețiile, setați Visual Studio ca implicit:



1.6 CLion MinGW – Optiunea 2 (mai rapidă, necesită consolă externă)

1.6.1 Eroare Clear în CLion

Exemplu:

A screenshot of the CLion Test console window. The tab bar at the top shows 'Run' and 'Test' (which is currently active). The console output area displays the following text:

```
C:\Users\ardel\CLionProjects\Test\cmake-build-debug\Test.exe
'clear' is not recognized as an internal or external command,
operable program or batch file.
```

The console has standard scroll and search controls on the left side.

Soluție:

Accesați fișierul main.cpp și derulați la liniile 113-117 în funcția displayGrid.

```

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("clear");
116 #endif

```

Modificați după cum urmează:

În ramura else de la linia 116

```
system("clear");
```

înlocuiți cu

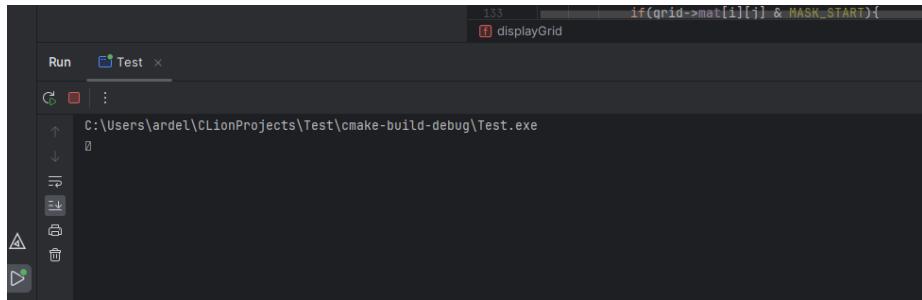
```
system("cls");
```

```

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("cls");
116 #endif
117    for(i=0; i<grid->rows; ++i)

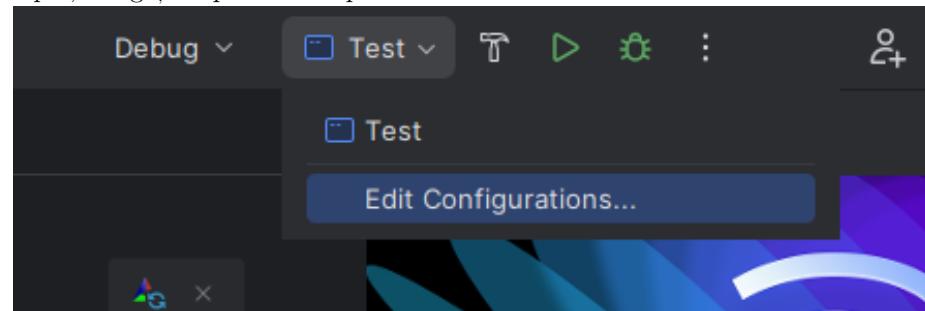
```

1.6.2 CLion nu afișează grila



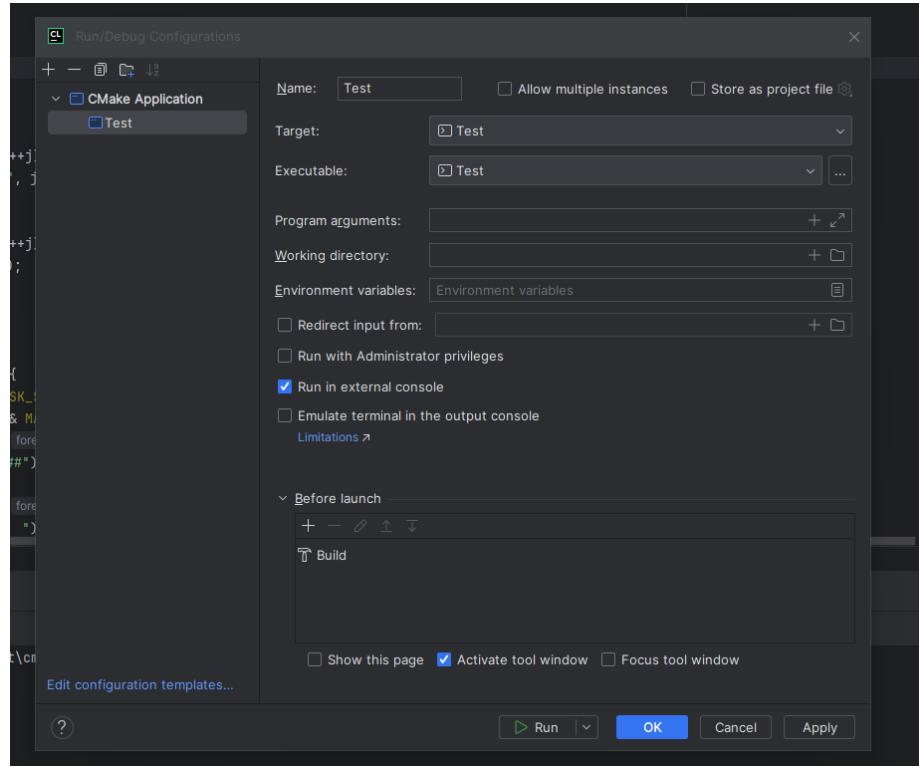
Copiați fișierul 'grid.txt' în folderul 'cmake-build-debug'.

Apoi, mergeți în partea dreaptă sus a ecranului:



Selectați 'Edit Configurations' și bifăți următoarele opțiuni:

- Rulați în consolă externă



1.7 Comanda de rulare pe Mac

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