

Pentru cei interesati sa foloseasca Visual Studio Code pentru laboratorul PC, aveti mai jos un scurt ghid de setare a mediului de lucru (structura fisiere, syntax error highlight, autosave etc).

Atentie - Din punct de vedere didactic este obligatoriu folosirea liniei de comanda pentru compilarea si executarea programelor C!

Nota: Folosirea VS code este optionala, la laborator se poate folosi orice editor de texte cu compilare si executie din linia de comanda. Idee este de a va face voua viata mai usoara pe partea de syntax highlight si formatare si de a va familiariza cu unul din editoarele larg utilizate în dezvoltarea software.

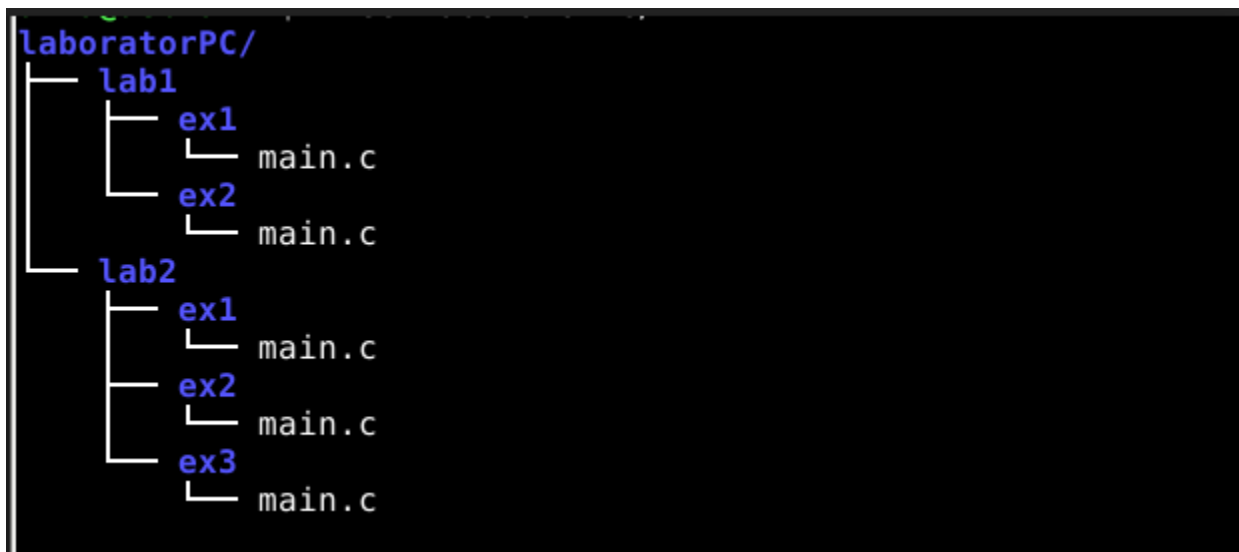
1. Structura mediu de lucru

Se recomanda crearea următoarei structuri de directoare / fișiere in directorul home al user-ului logat (exemplu: /home/alex)

LaboratorPC

- {folderLaborator}
 - {folderExecitiuNr}
 - main.c (aici vom scrie codul)

Exemplu:



Idee este sa setam workspace-ul VS Code in folder-ul laboratorPC astfel incat sa nu refacem setup-ul pentru fiecare laborator.

De asemenea dorim sa ne fie usor sa navigam catre exercitiile rezolvate anterior pentru a ne improspata memoria.

Crearea unui folder per exercitiu pare a fi redunanta, dar vom vedea mai tarziu ca un program C poate avea mai mult fisiere. Prin urmare este indicat sa folosim aceasta structura de la bun inceput.

Prin conventie am denumit numele fisierului ce contine codul "main.c", deoarece acest fisier contine functia main(). Insa numele main.c nu este obligatoriu, programul va rula ok si cu alt nume. Folosirea acestei conventii poate fi utila mai tarziu cand vom avea mai multe fisiere.

Daca se doreste numele directorului pentru exercitiu poate fi mai sugestiv. De exemplu "maxim_numar_ex1"

De exemplu:

Cand veti lucra la exercitiu 3 laborator 2, veti scrie codul in

`~/laboratorPC/lab2/ex3/main.c`

Pentru compilare si rulare din terminal vom rula

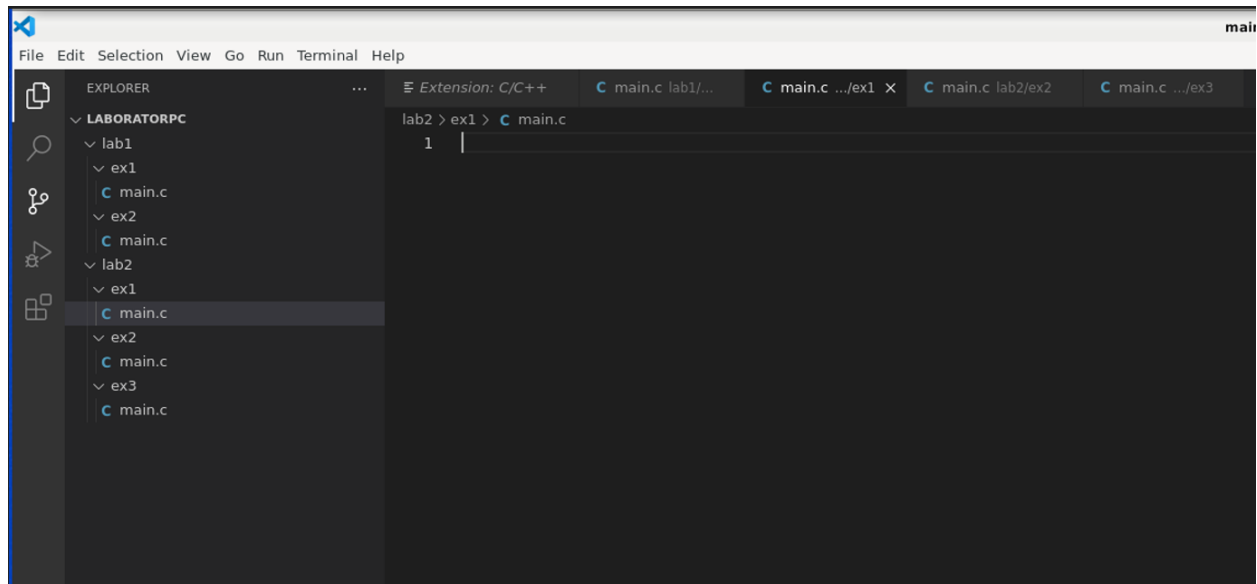
```
cd ~/laboratorPC/lab2/ex3 // o singura data pentru ajunge in locatia dorita
gcc -Wall -o main main.c
./main
```

2. Deschidem VS Code.

Din terminal navigam in directorul creat ***laboratorPC*** si rulam ***code .*** (*code spatiu punct*)

Folderul se poate deschide si din VS code direct: ***File -> Open Folder..***

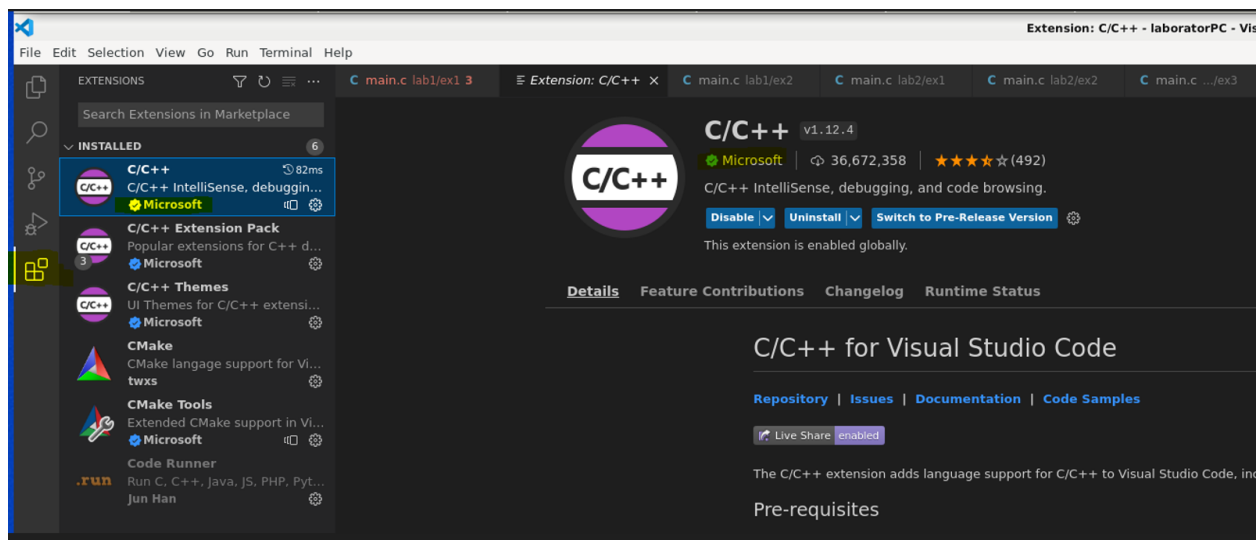
Ar trebui sa vedem urmatoare structura a mediului de lucru.



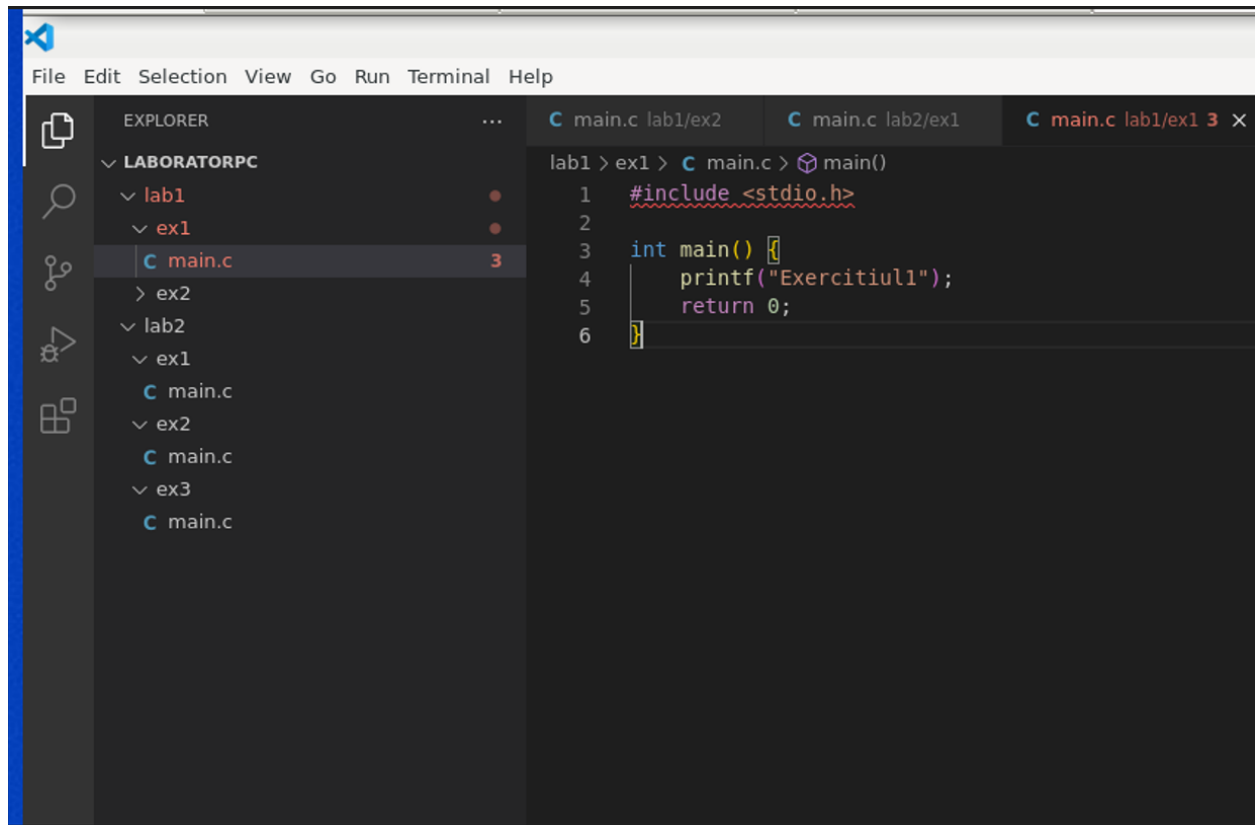
3. Ne asiguram ca avem instalat plugin C/C++

In mod normal vine pre-instalat.

- **VS Code -> Extensions** (panel stanga, highlight cu galben mai jos) sau **CTRL + Shift + X**
- Cautam C/C++ si ne asiguram ca plugin-ul de la Microsoft e instalat.



4. Scriem un program de test in unul din fisierele main.c



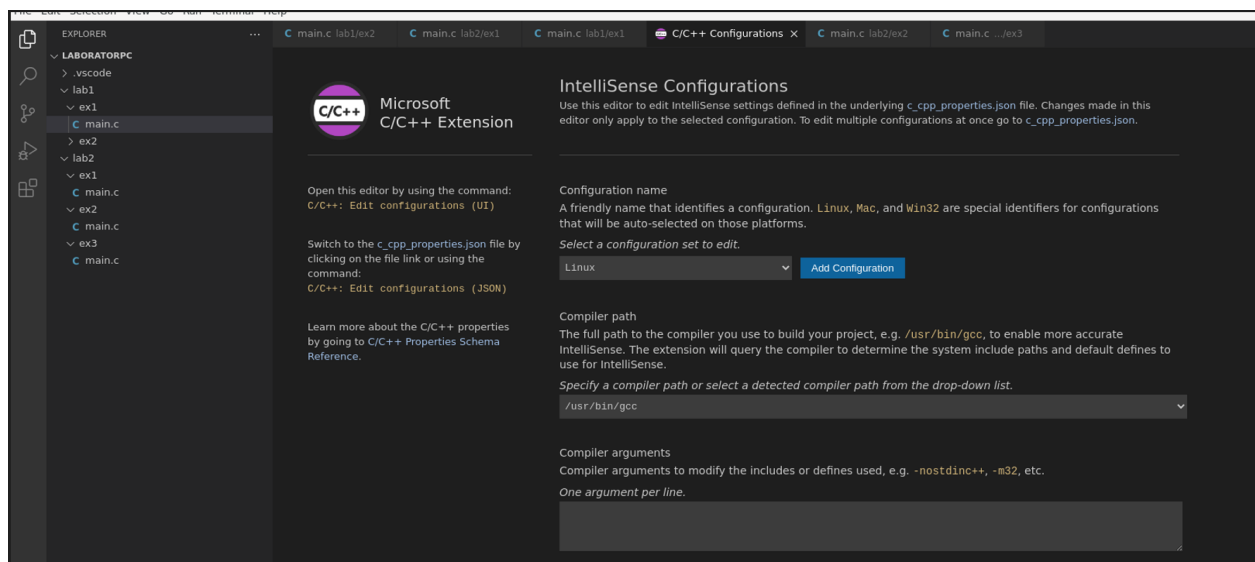
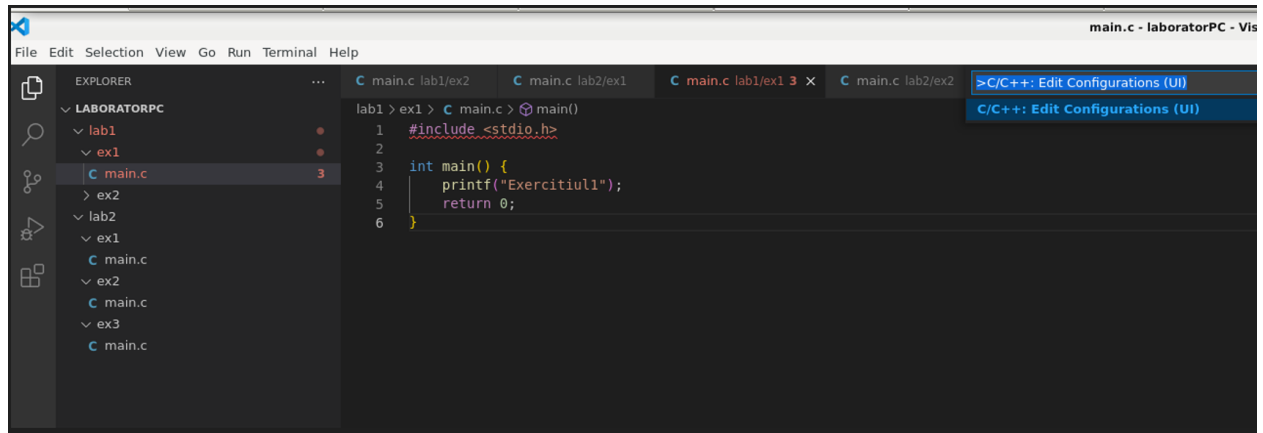
Observam ca importul (**#include ...**) este subliniat cu rosu.

Orice apare subliniat cu rosu, Intellisense-ul de la VS Code ne spune ca este o eroare de compilare. Evidentierea erorilor de compilare este de foarte mare ajutor si unul prin principalele motive pentru care folosim VS code. Incercati sa va formati un reflex in a observa ce este subliniat cu rosu.

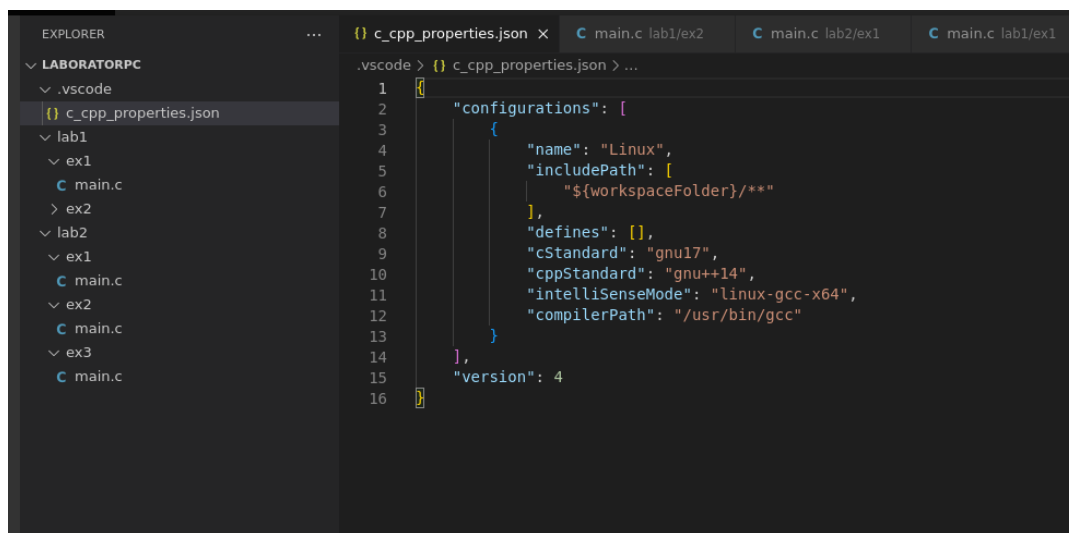
In cazul de fata nu este vorba de o eroare de compilare, ci faptul ca Intellisense nu este configurator corect.

5. Setare Compiler Path pentru C/C++ intellisense.

- CTRL + Shift + P - Deschide terminal rulare comenzi VS
- Search for C/C++: *Edit Configurations (UI)*
- Compiler Path -> select /usr/bin/gcc
- Inchideti tab-ul de configurari
- Navigati catre main.c -> eroare ar trebui sa dispara.



A se observa ca VS code a creat un nou folder `.vscode` si un nou fisier `c_cpp_properties.json` pentru salva configurariile cerute mai sus

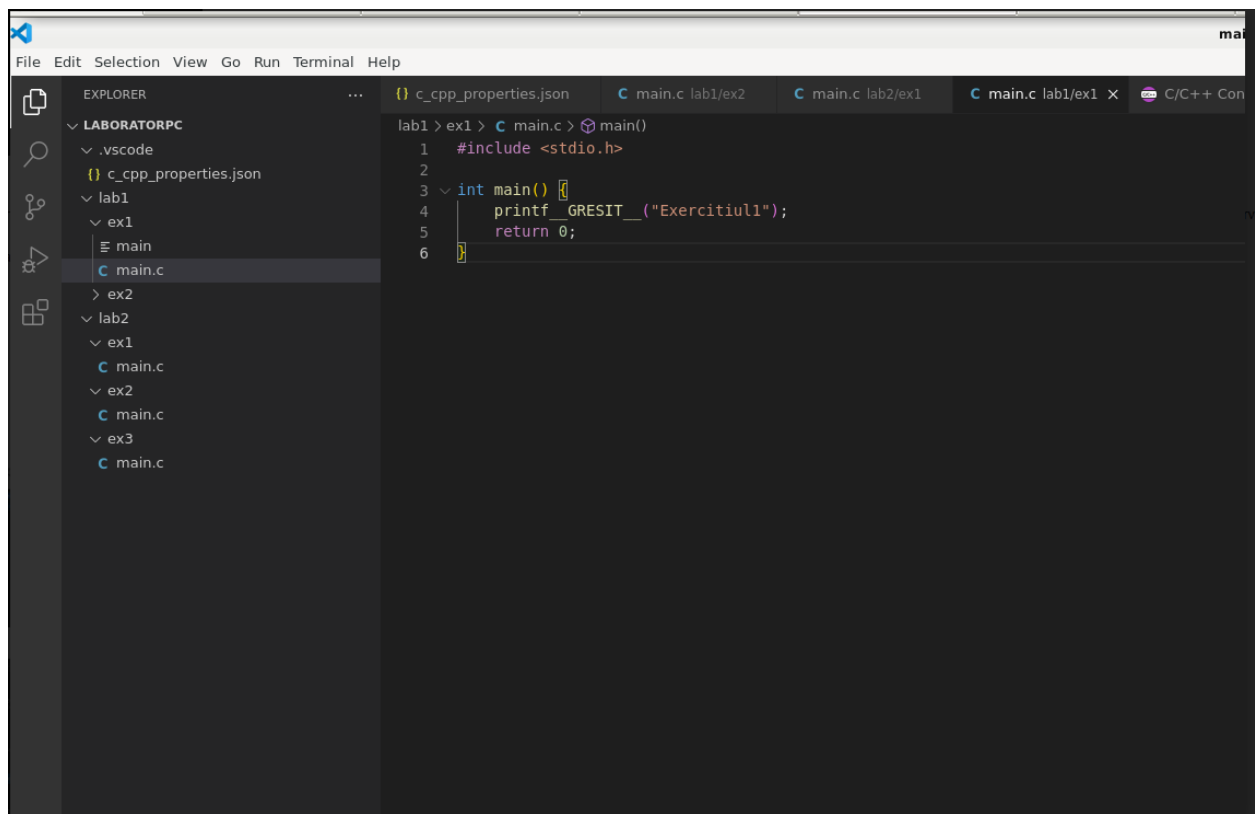


6. Enable auto code analysis.

By default - Pentru a avea highlight la erorile de compilare este necesar sa rulam task-ul de build.

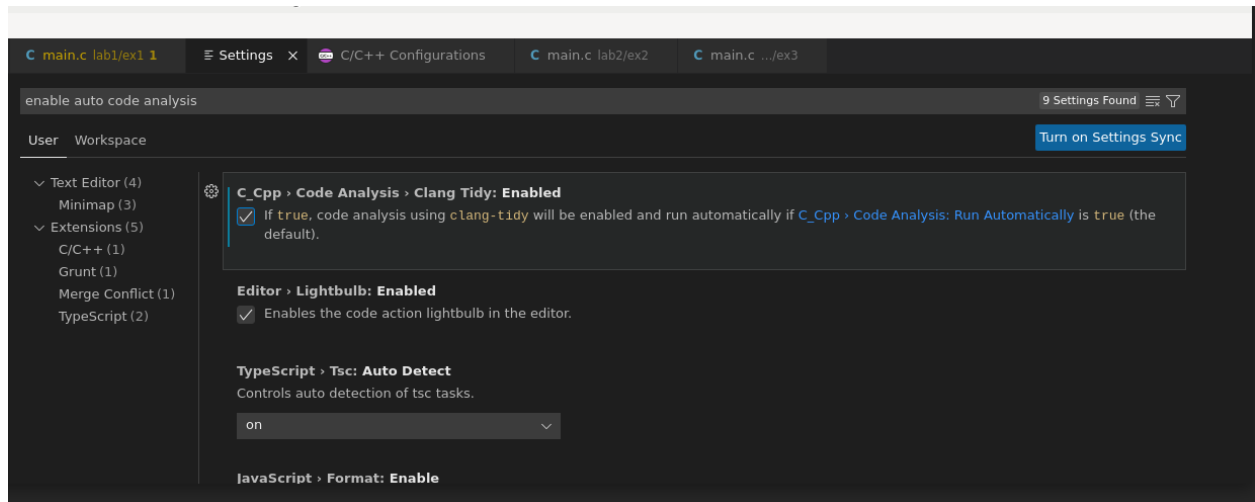
Noi dorim sa avem evidentierea erorilor în timp real.

Daca gresim ceva intentionat in program, observam ca VS code nu ne face highlight (functia printf scrisa gresit):



Pentru a rezolva acest lucru trebuie sa facem următoarele:

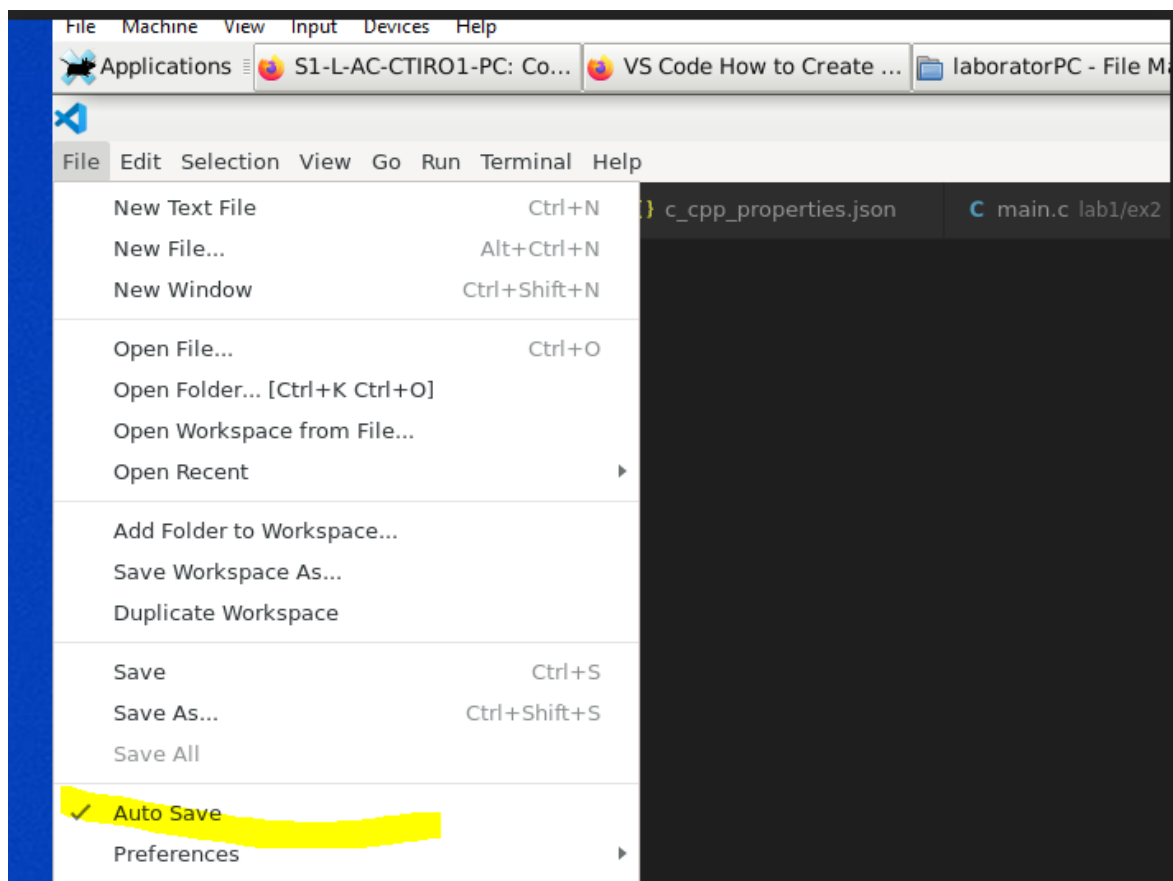
- **File -> Preferences -> Settings** <-> Sau **CTRL + ,** (CTRL Virgula)
- Search **enable auto code analysis**
- Marcheaza **enabled** checkbox-ul



Comenzile de mai sus au facut ca analiza sa ruleze automat **la salvarea fisierului**.

Pentru a nu fi necesar sa salvam fisierul de fiecare data cand scriu ceva vom face enable la autosave.

- **File -> Autosave**



7. Formatare cod si alte comenzi utile

7.1 Formatare cod:

Este necesara sa avem o fomatare buna a codului pentru a spori lizibilitatea. Acest lucru va va ajut atat pe voi cat si pe altii ce se uita la codul vostru.

VS code ne vine in ajutor si ne ajuta sa formatam automat codul cu ajutorul urmatoarelor comenzi:

- Select code lines (or **CTRL + A** to select entire file)
- **CTRL + K, CTRL F** (se poate tine apasat CTRL urmat de apasarea K si apoi F)

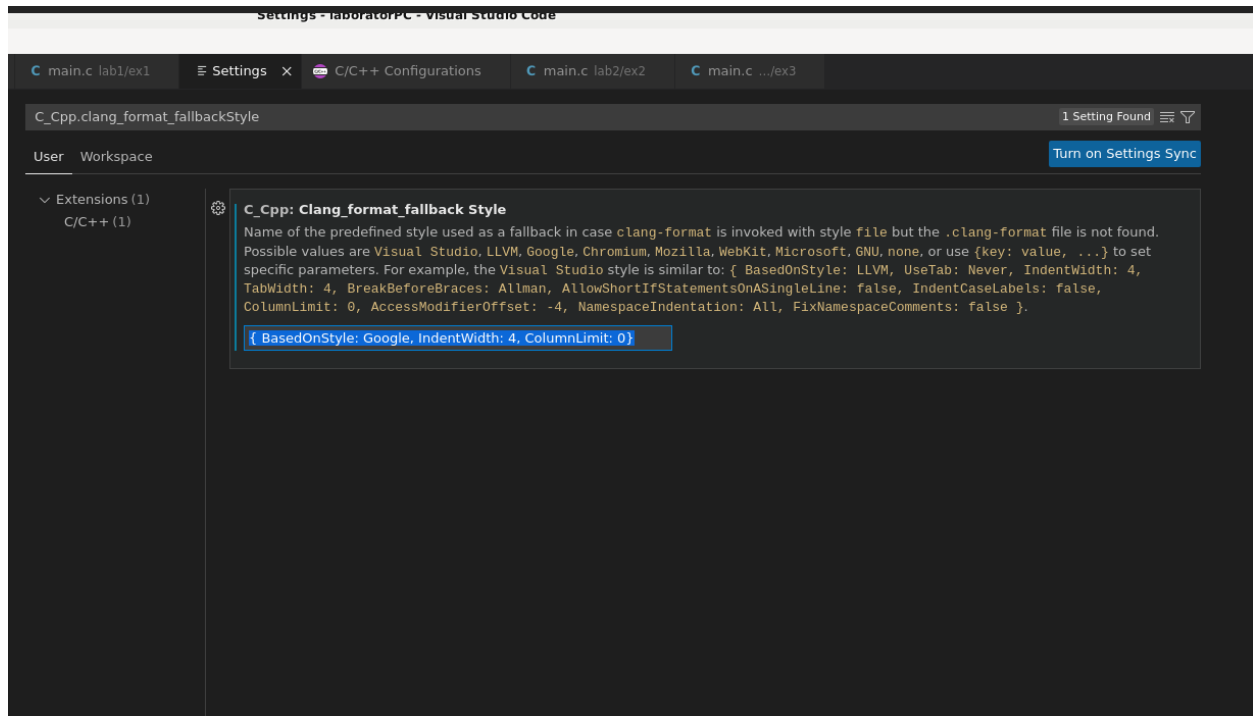
7.2 Setare format acolade {}

A se observa ca la punctul 7.1, dupa formatarea codului acolada de deschidere a unul bloc sau functie a fost mutata din rand nou (stil visual studio).

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Exercitiul1");
6      return 0;
7  }
```

Pentru modifica stilul de formatare si a ne alinia cu cel folosit in laborator (si mai tarziu la OOP), putem sa facem urmatoarele

- **File -> Preferences -> Settings**
- search for **C_Cpp.clang_format_fallbackStyle**
- Inseram **{ BasedOnStyle: Google, IndentWidth: 4, ColumnLimit: 0 }**



Dupa setarea anterioare, mergem inapoi in fisierul cod si rulam

1. **CTRL A**
2. **CTRL K, CTRL F**

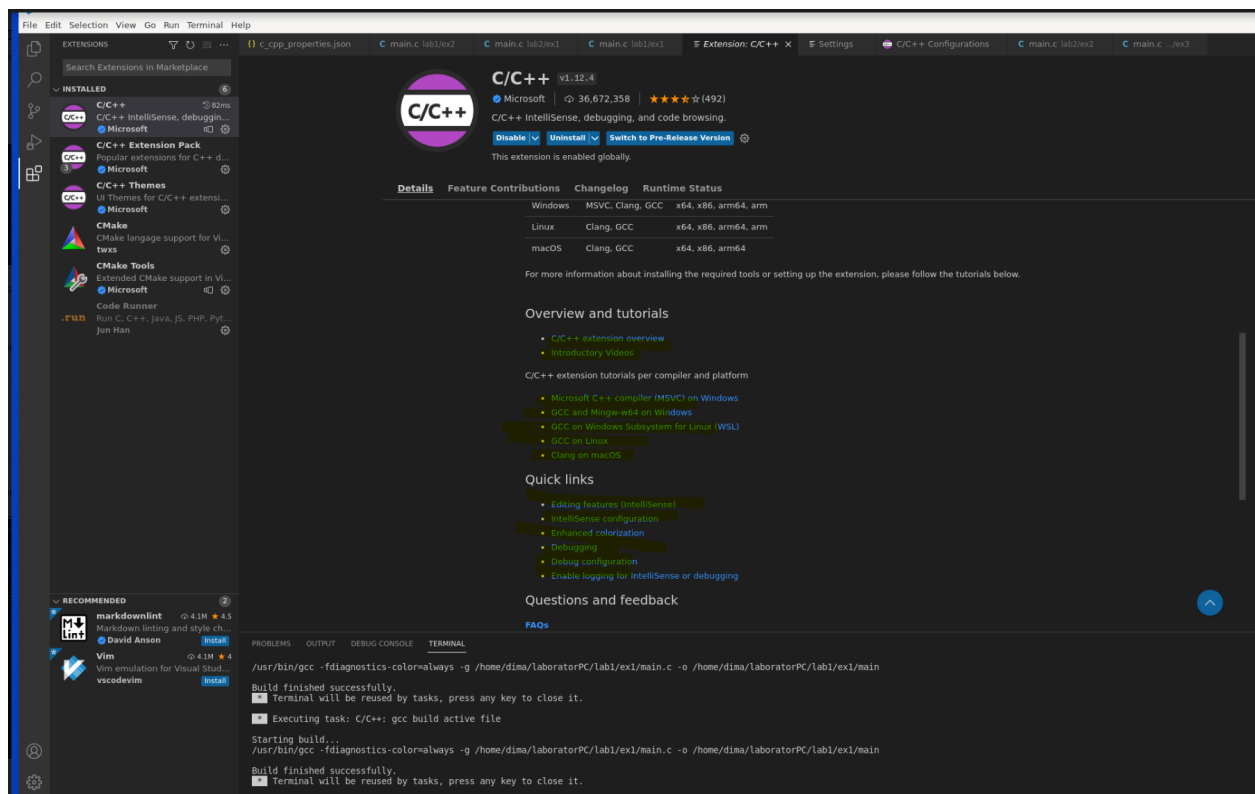
7.3 Comment /Uncomment code

- Selecteaza cod (altfel se va aplica la linia unde avem cursorul)
- **CTRL + /**
- Pentru uncomment rulăm același set de comenzi.

SAU

- select lines -> CTRL + K -> CTRL C // coment selected lines
- select lines -> CTRL + K -> CTRL U // uncomment selected lines

Pentru mai multe informatii folositi cu incredere link-urile de pe pagina plugin-ului (CTRL + Shift + X -> C/C++)



Salu pagina oficiala VS Code:

<https://code.visualstudio.com/docs/getstarted/introvideos>

Mai multe extensii(plugin) VS Code:

<https://code.visualstudio.com/docs/editor/extension-marketplace#:~:text=VS%20Code%20extensions%20let%20you,APIs%20used%20by%20VS%20Code.>