INSTALLAZIONE IDE CODEBLOCKS VER. 20.03 SU WIN 11 (anno 2022)

(per MacOS è disponibile la versione CodeBlock 25.03 qui:
http://www.codeblocks.org/downloads/binaries/)

Alcune informazioni iniziali:

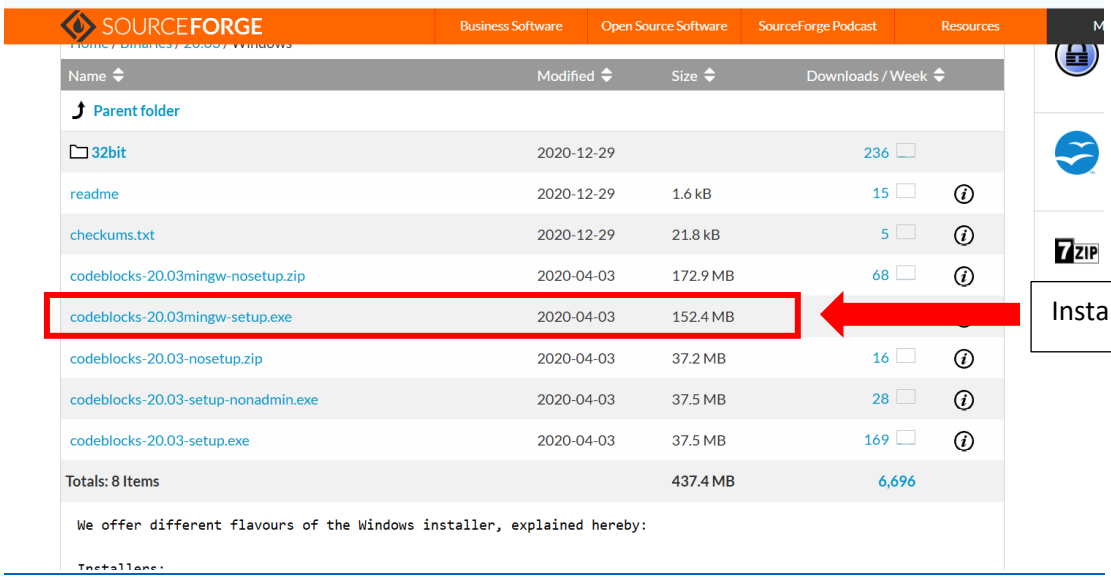
> Quale è l'ultima versione del linguaggio C? C23 è il nome informale, nome: ISO/IEC 9899/2024.
>
> Quale è l'ultima versione dell'IDE CodeBlocks? CodeBlocks 25.03, ma installiamo la 20.03 che non da problemi di sicuro.

Cosa è CodeBlocks? Un IDE (Integrated Developement Enviroment)

Cosa è GCC? GNU Compiler Collection, è un compilatore nato per compilare codice in linguaggio C su piattaforme LINUX, poi si è sviluppato ed ora è un compilatore multilinguaggio e multipiattaforma. Consente di compilare C, JAVA, C++, Fortrand, ADA per diverse piattaforme come GNU/Linux (SO realizzato con software libero), Windows e altre architetture.

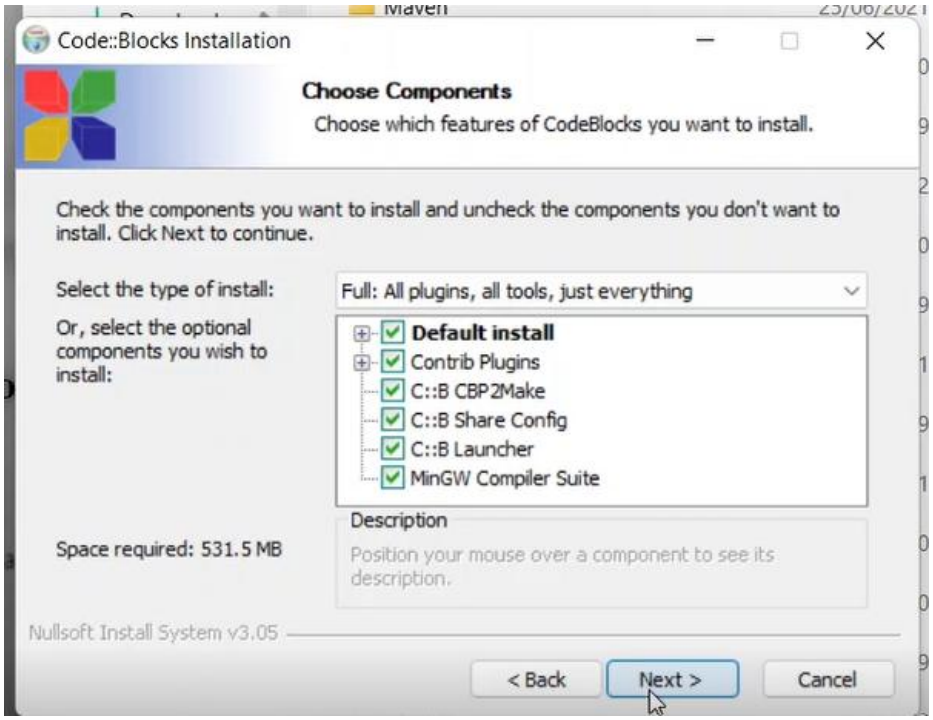
Cosa è MingW? Ambiente di compilazione per supportare GCC su ambienti Windows

Detto questo, su un sistema Windows dobbiamo installare CodeBlocks 20.03 con MingW da qui:
https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/

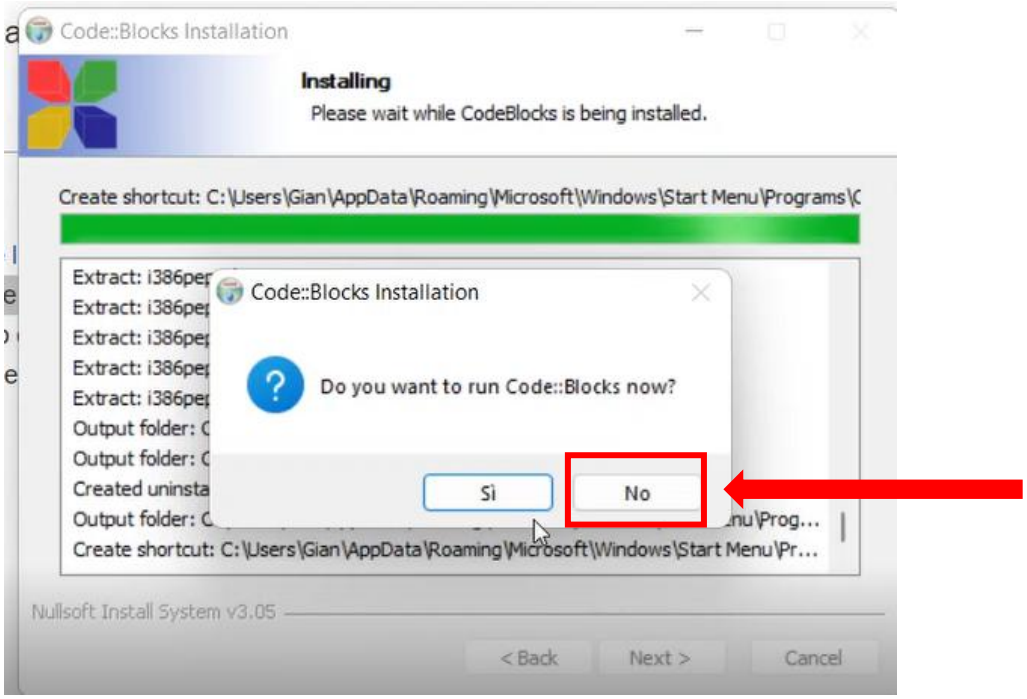


Installiamo questo

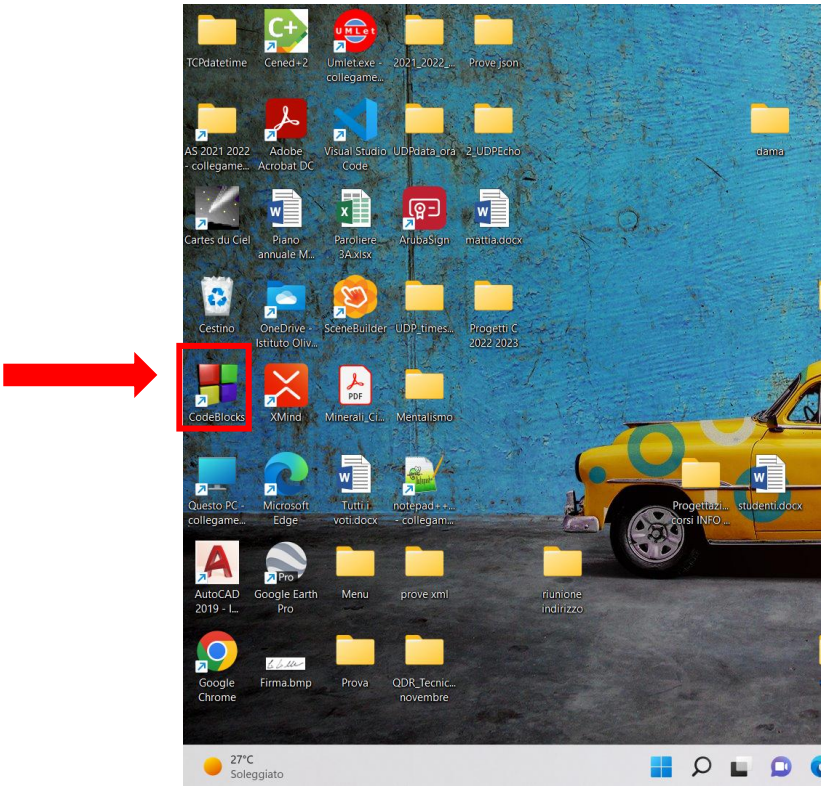
Installiamo tutto (531 MB)



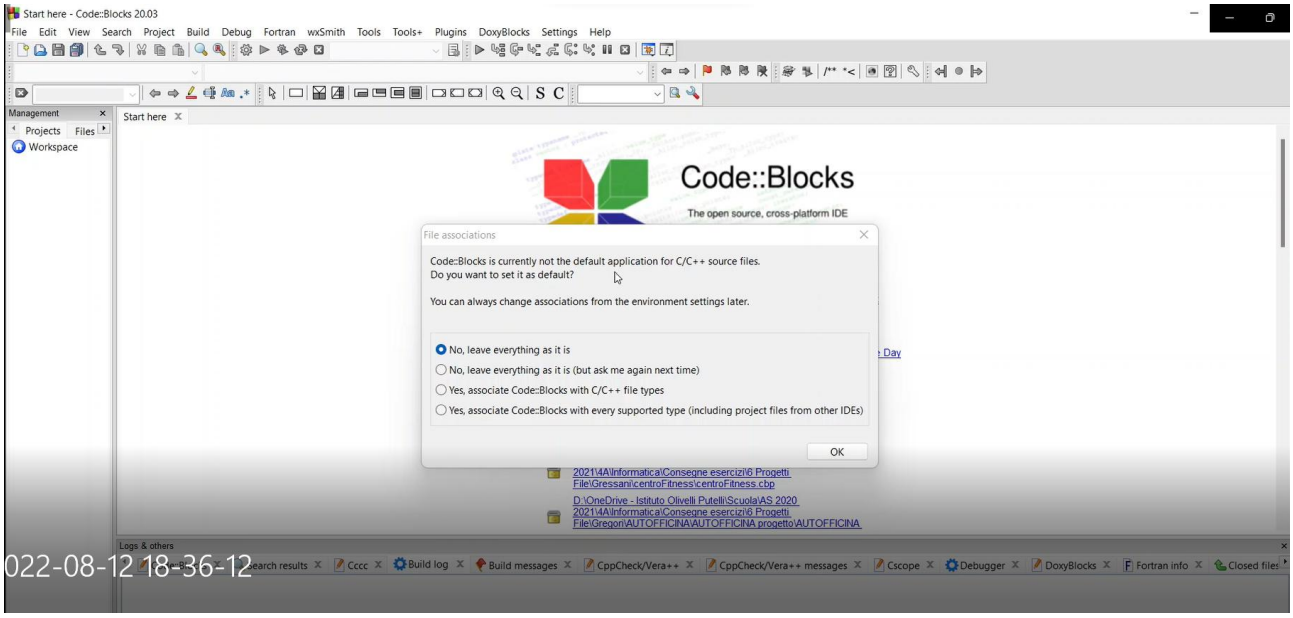
Non lo lanciamo subito



Lo lanciamo dall'icona sul desktop

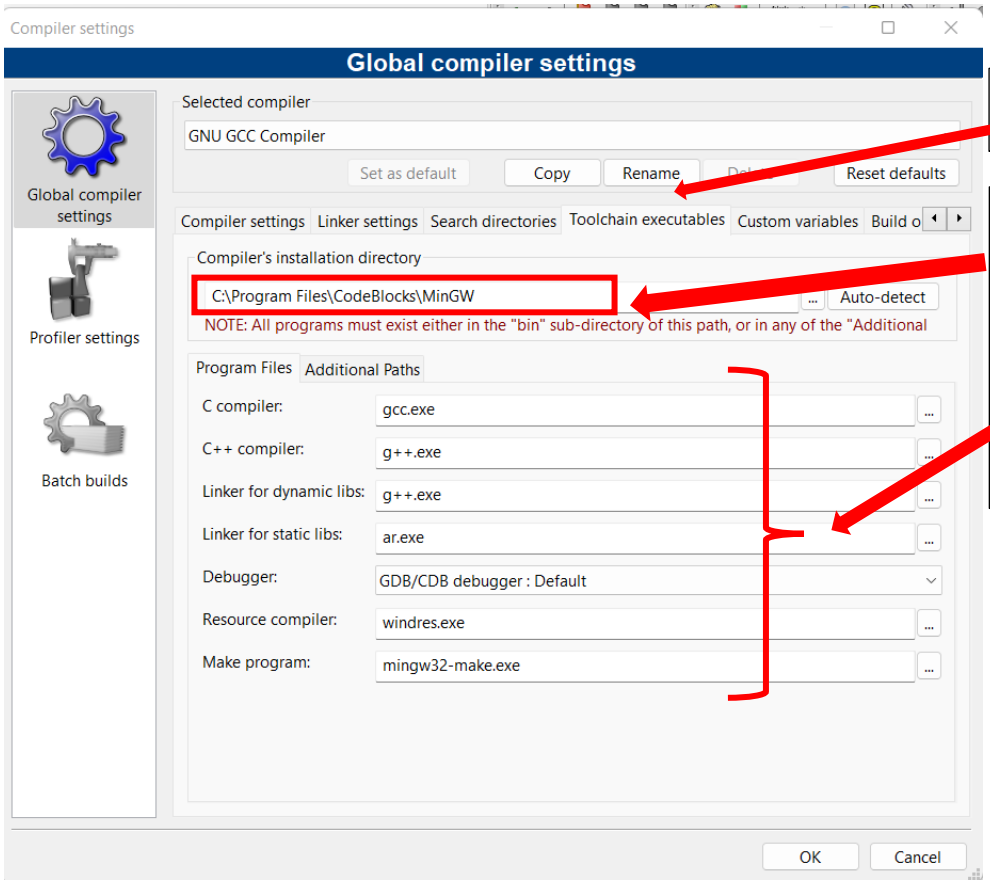


Lasciamo impostazioni di default



Verifichiamo che le impostazioni siano le seguenti

menu: Settings/Compiler/Compiler setting/Toolchain executables



Toolchain executables

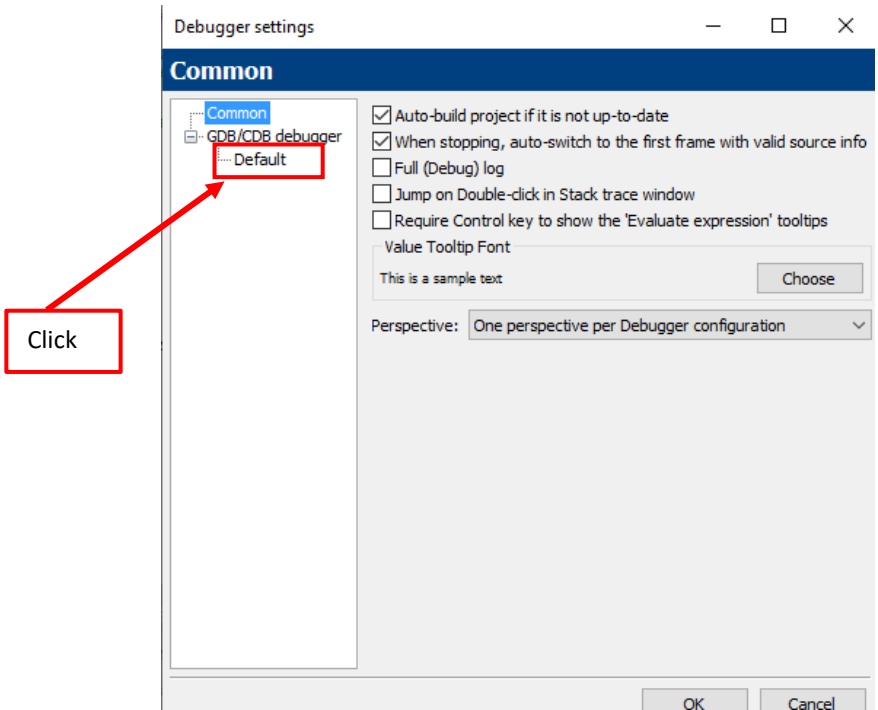
Qui deve essere indicato il path di installazione.

La cartella MinGW deve contenere la cartella “bin”

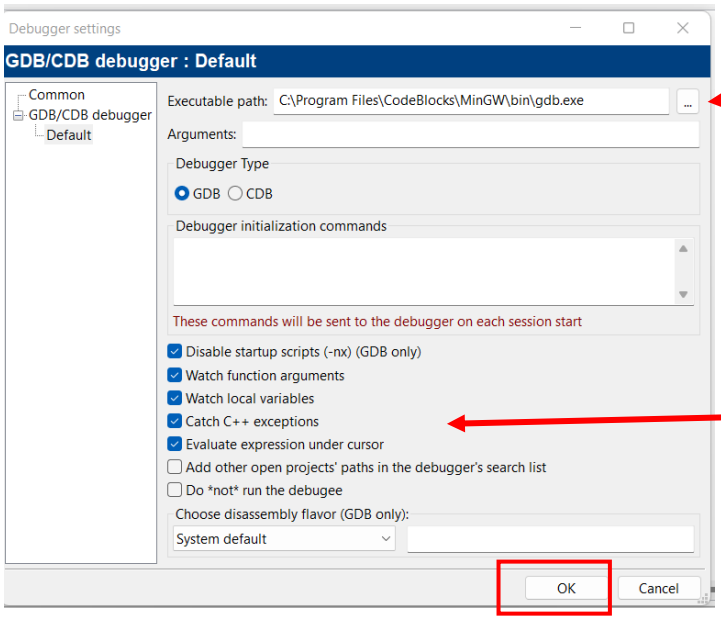
La cartella “bin” deve contenere questi file .exe

Configuriamo il debugger (fondamentale)

Menu:Settings/debugger…



Selezioniamo il path corretto che contenga il file gdb.exe



Cliccando su … selezionare il file "**gdb.exe**" che si trova nel percorso **CodeBlocks\MinGW\bin** nella cartella in cui è installato CodeBlocks (generalmente la cartella in cui è installato è C:\Program Files)

Spuntare le opzioni come nell'immagine

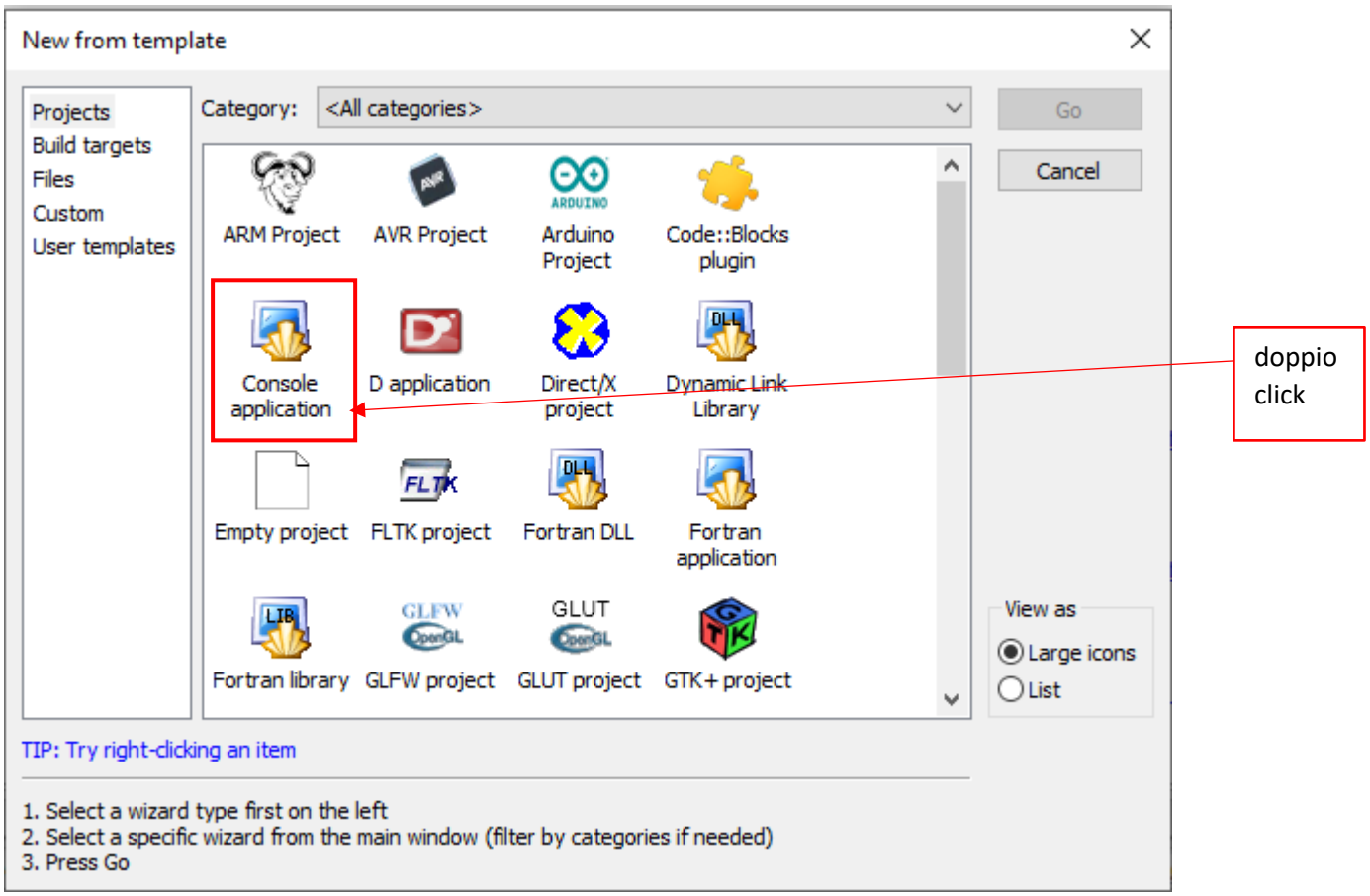
Click

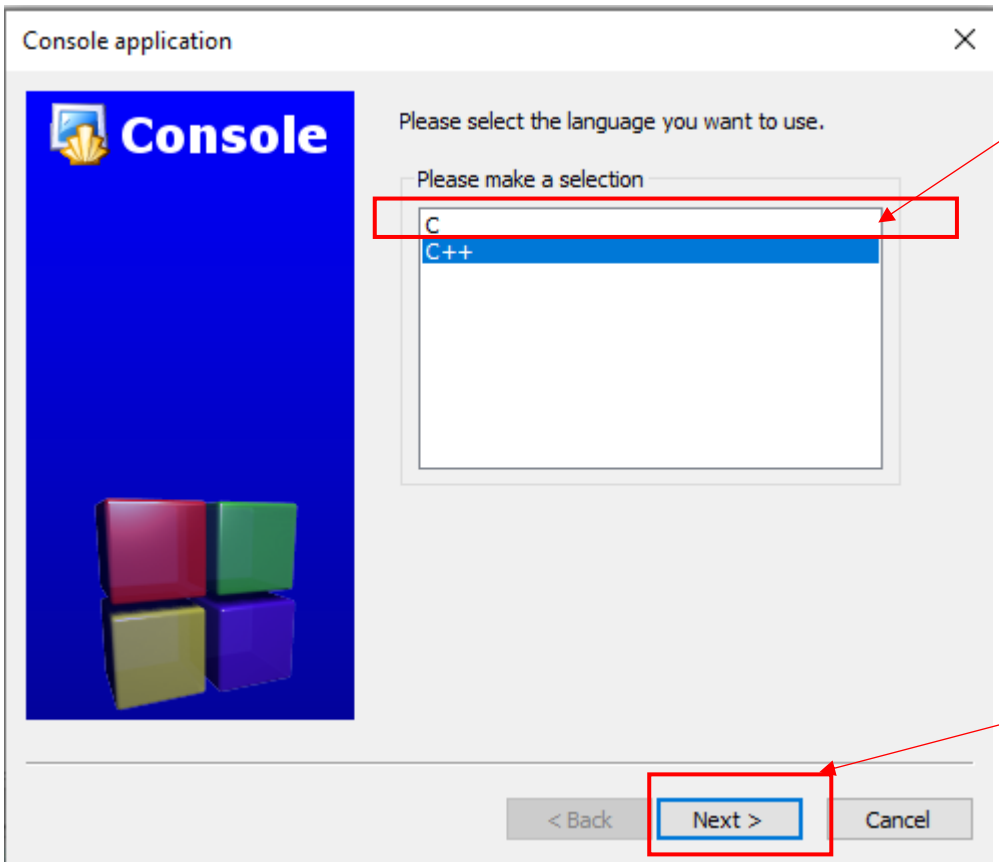
Primo progetto

Una volta scaricato, installato, avviato ed attivato e configurato l' IDE iniziamo creando una applicazione Console in CPP (le applicazioni Console sono quelle che realizzeremo quest'anno, sono senza interfaccia grafica)

menu: file/new/project

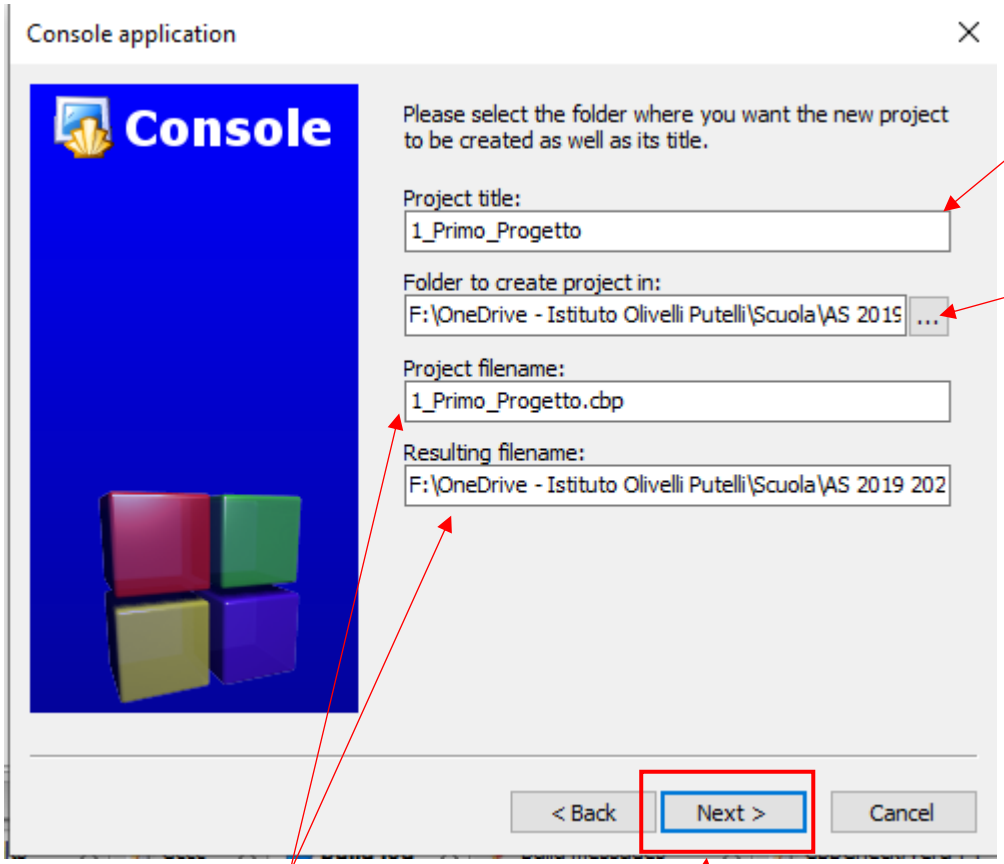
appariranno le seguenti finestre questa finestra, seguiamo le indicazioni nelle immagini.





Selezioniamo C

click



Indichiamo il nome del progetto

cliccando su "…" selezioniamo la cartella in cui vogliamo salvare i nostri progetti:

Ognuno crei una apposita cartella chiamata con il proprio cognome

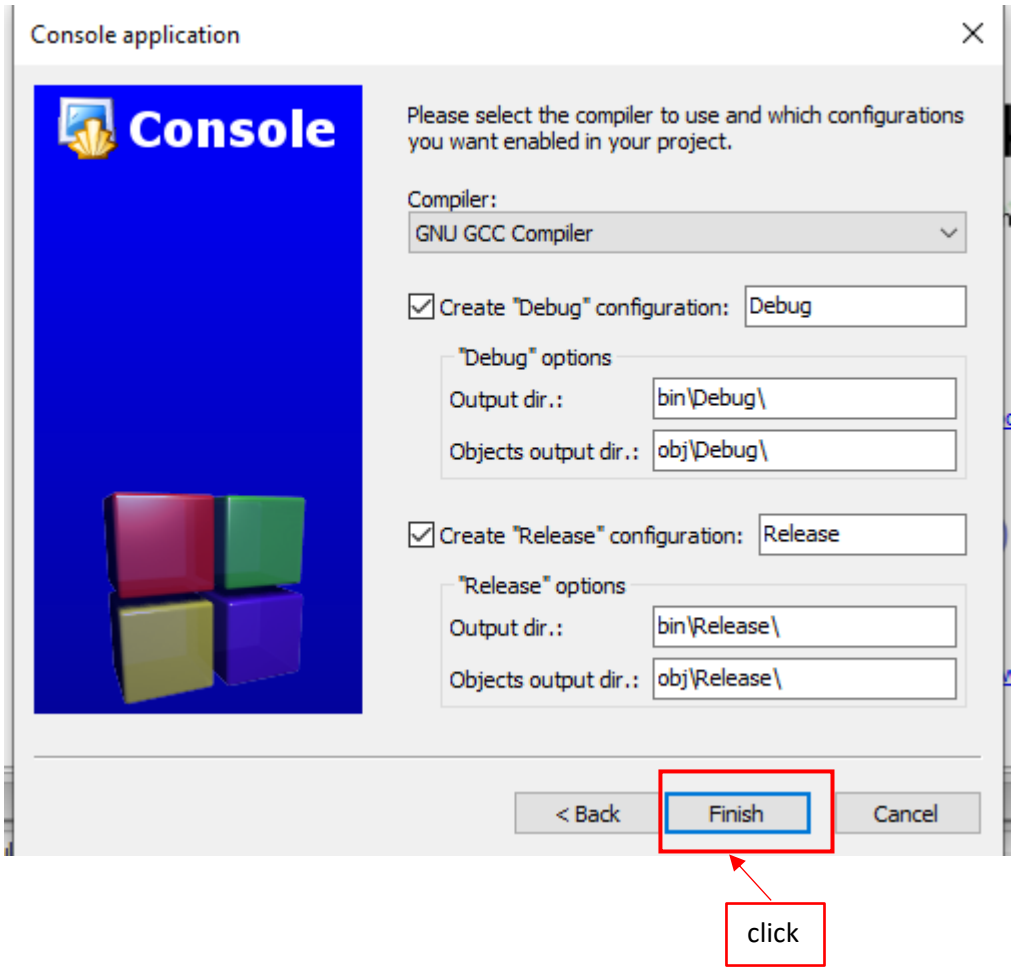
Cognome_Progetti_C

nella propria cartella personale.

ATTENZIONE: poiché l'IDE è utilizzato anche da altre classi, sarà necessario selezionare la propria cartella che contiene i progetti ogni volta che si accede all' IDE! Chiaro?!

questi si compilano automaticamente

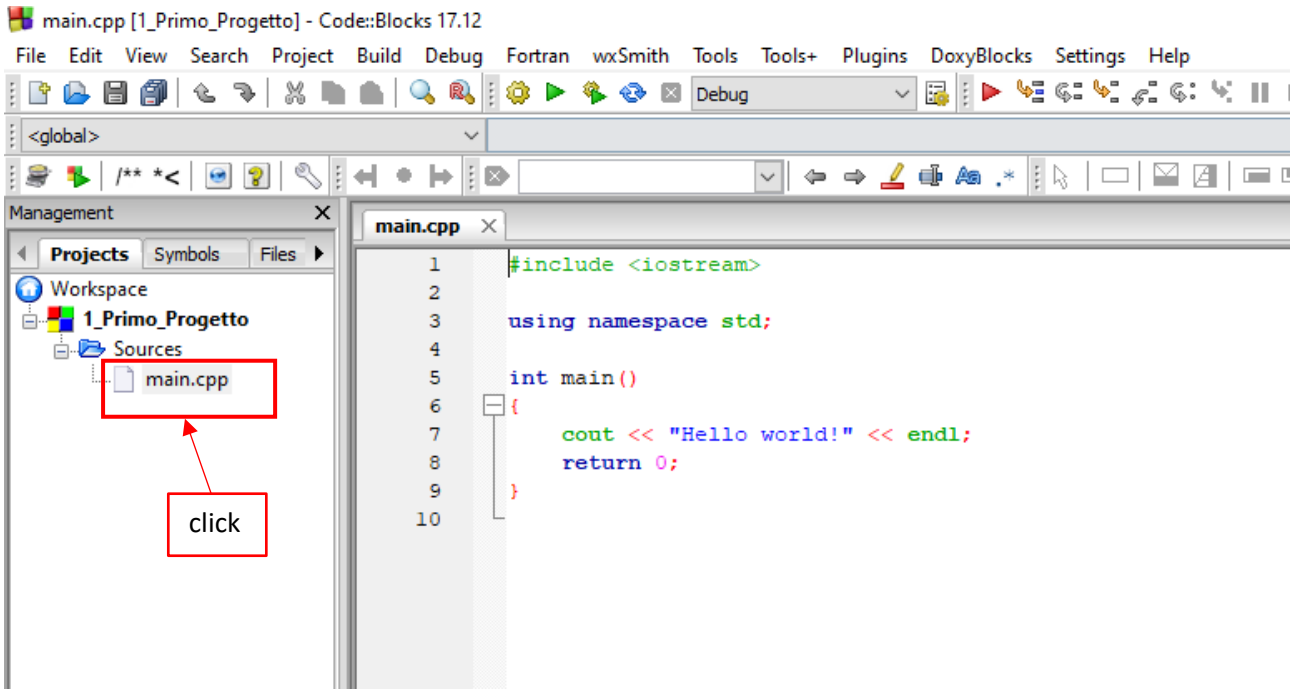
click



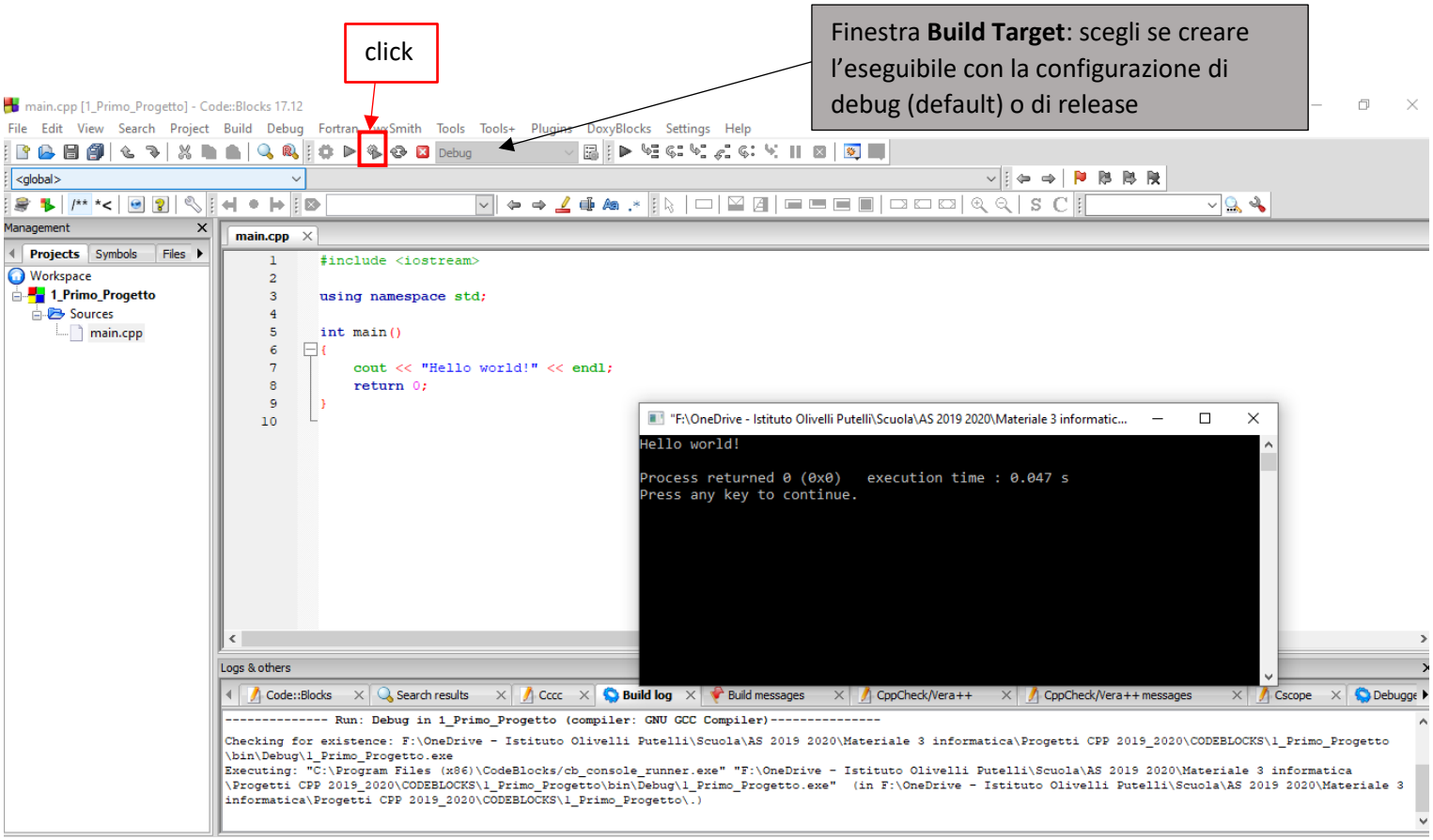
Questi flag danno la possibilità di scegliere, in seguito durante fase di compilazione, se creare l'eseguibile in configurazione debug (codice non ottimizzato ma compilazione più veloce) oppure in configurazione release (codice ottimizzato, eseguibile più pesante ma compilazione più lenta).

La scelta si fa successivamente, prima di compilare nella select box **Build target** (vedi dopo)

Il primo semplice programma ("Hello World") viene creato automaticamente. Visualizziamo il codice cliccando sul file "main.cpp" nella scheda "Projects" della finestra "Management" a sinistra

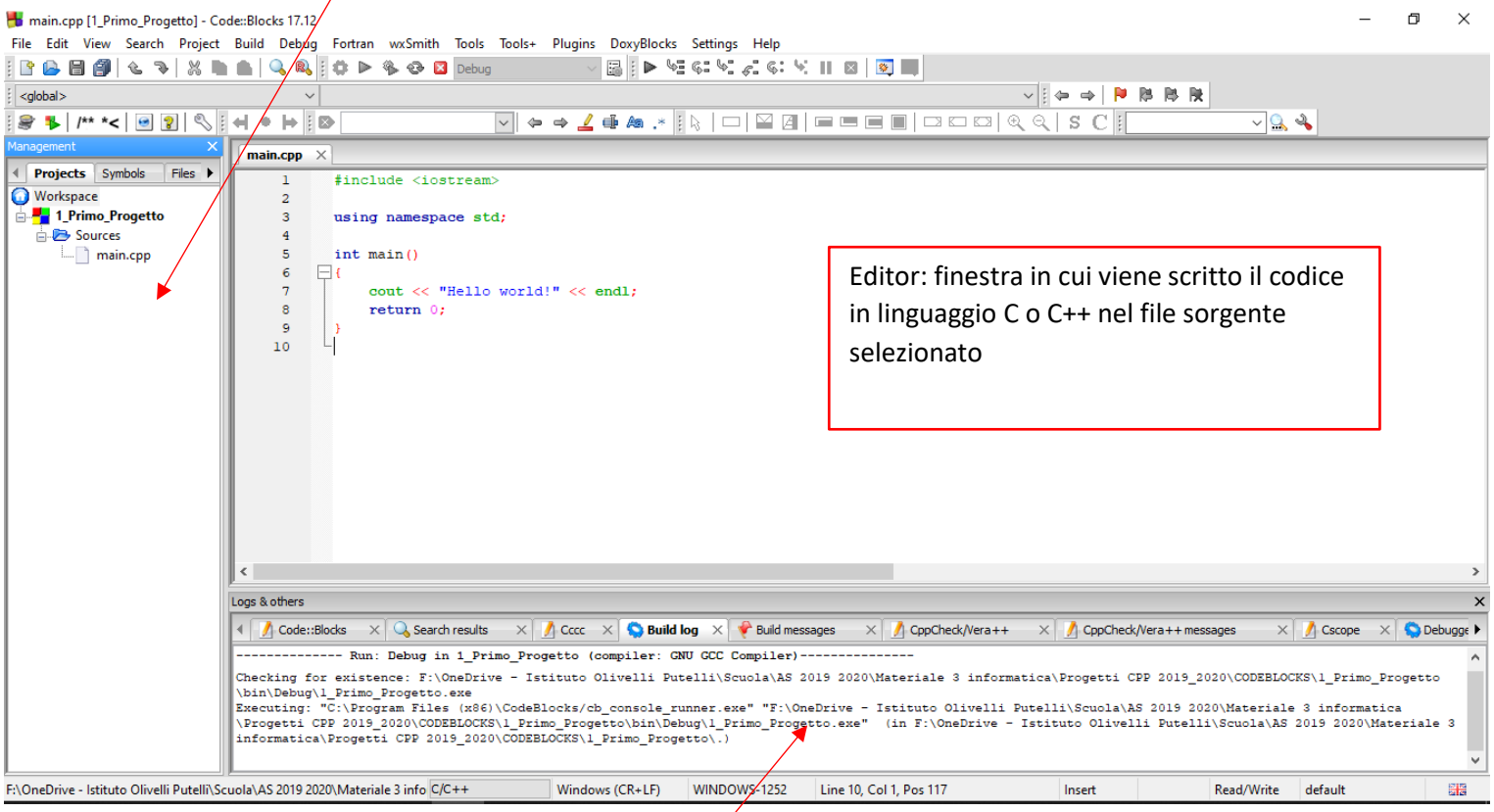


Per compilare, linkare ed eseguire il programma clicchiamo sul pulsante “Build and run” (freccia verde con ingranaggio. Si aprirà la console di Windows nella quale vedremo l’esecuzione).



Le finestre dell' IDE CodeBlocks

Management: mostra la struttura **logica** del progetto, ossia i nomi e le cartelle dei vari file sorgente che costituiscono il codice del progetto. All'inizio lavoreremo con un solo file sorgente chiamato main.cpp. Il file sorgente selezionato viene aperto e visualizzato nella finestra di Editor.



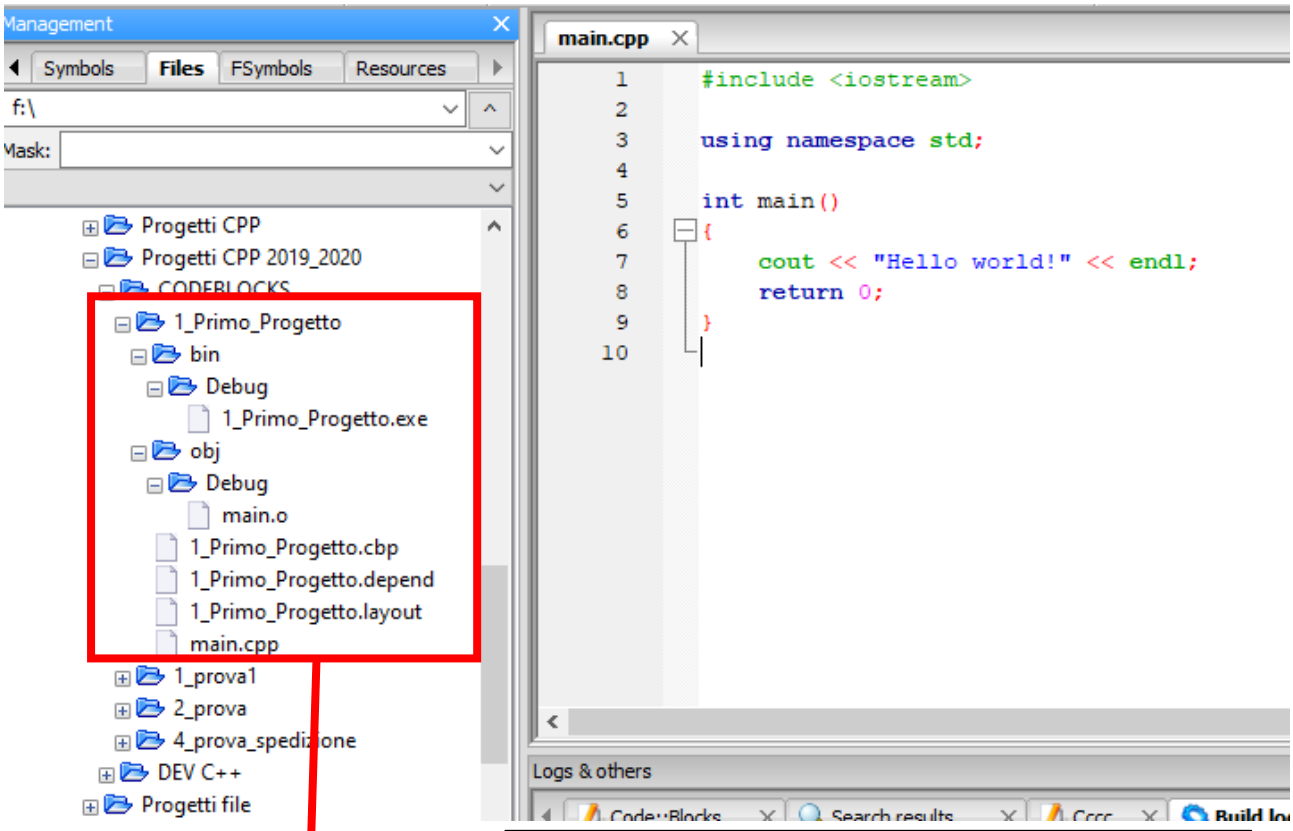
Editor: finestra in cui viene scritto il codice in linguaggio C o C++ nel file sorgente selezionato

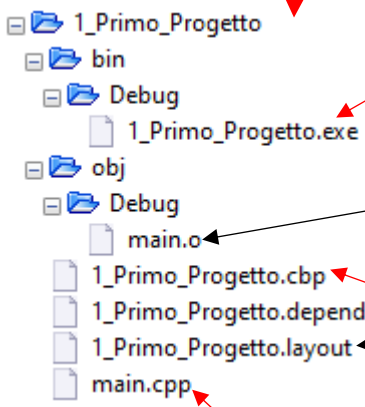
Logs & others: finestra che mostra eventuali messaggi di errore (o di successo) avvenuti durante la compilazione o l'esecuzione del codice sorgente.

I file del progetto

Quando viene creato un nuovo progetto, anche vuoto, automaticamente l' IDE crea all'interno di una cartella con il nome del progetto, una serie di cartelle e file, vediamo cosa sono.

Per visualizzare tutti i file del progetto si può aprire direttamente la cartella di progetto con il visualizzatore del file system di windows, oppure accedere alla scheda "files" della finestra Manegement:





File eseguibile. E' il file in linguaggio macchina (LM). E' il risultato della compilazione e del linking. Portabile ed eseguibile su ogni piattaforma per il quale è stato compilato (nel nostro caso su ogni Windows 10 a 64 bit). Si genera automaticamente con il pulsante "Build" (o "Build and run") dell' IDE.

File oggetto. E' il file che si ottiene dalla compilazione. Si genera automaticamente quando si compila il file sorgente

Questi tre file file sono creati dall'IDE CodeBlock. Non contengono codice, sono file "di servizio" per la gestione del progetto. Importante è il file con estensione .cbp perché **è quello che dobbiamo cliccare per aprire il progetto con l'IDE CodeBlock**

I file con estensione ".c" oppure ".cpp" (in questo caso è solo uno) sono i file che ci interessano di più. **Sono i file sorgente, quelli che contengono il codice scritto da noi in C o C++. Se scegliamo di realizzare un progetto in linguaggio C l'estensione sarà .c**

PROBLEMI CON IL DEBUGGER?

Il debugger potrebbe non funzionare e mostrare il seguente messaggio di errore

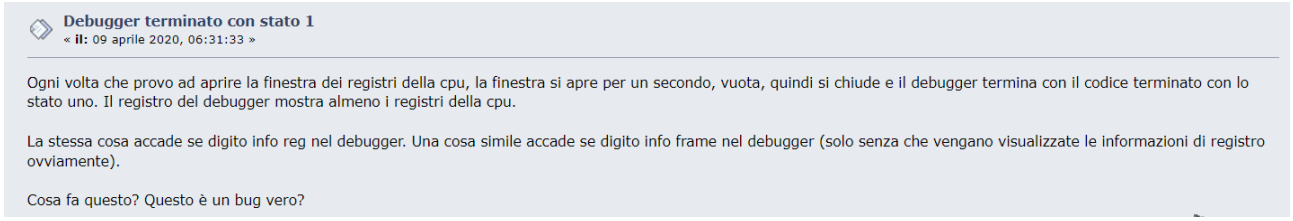
“Starting the debugger failed: No executable specified, use target exec”

Il problema è nel pathname del progetto. E’ necessario che il pathname non abbia spazi e o caratteri speciali, prova a spostare il progetto IN UNA CARTELLA IN CUI TUTTO IL PATHNAME non ha alcuno spazio.

ALTRI PROBLEMI CON IL DEBUGGER?

Il debugger potrebbe non funzionare correttamente e (non si ferma ai breakpoint) e termina con status=1

soluzioni suggerite qui: https://forums.codeblocks.org/index.php?topic=23787.0



Debugger terminato con stato 1
« **il:** 09 aprile 2020, 06:31:33 »

Ogni volta che provo ad aprire la finestra dei registri della cpu, la finestra si apre per un secondo, vuota, quindi si chiude e il debugger termina con il codice terminato con lo stato uno. Il registro del debugger mostra almeno i registri della cpu.

La stessa cosa accade se digito info reg nel debugger. Una cosa simile accade se digito info frame nel debugger (solo senza che vengano visualizzate le informazioni di registro ovviamente).

Cosa fa questo? Questo è un bug vero?

soluzione 1:



molti
Nuovo arrivato a distacco singolo
Post: 7

Re: Debugger terminato con stato 1
« **Risposta #11 il:** 09 aprile 2020, 12:44:20 »

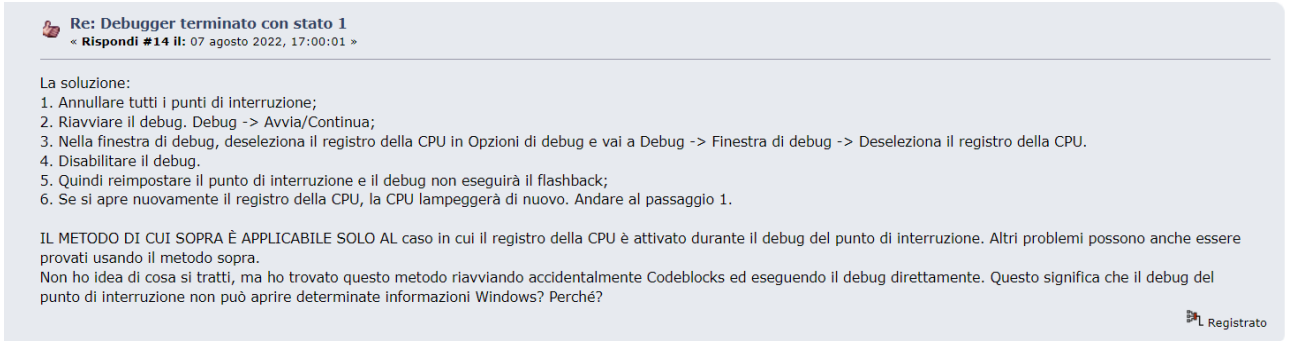
Sostituito gdb.exe con uno che ho trovato in una cartella di installazione di mingw 17.1 che ho trovato e ora funziona. Belle facce felici.

Peccato che ora sono troppo esausto per programmare.

Grazie per tutto il tuo aiuto!

Registrato

soluzione 2:



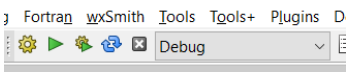
Re: Debugger terminato con stato 1
« **Rispondi #14 il:** 07 agosto 2022, 17:00:01 »

La soluzione:
1. Annullare tutti i punti di interruzione;
2. Riavviare il debug. Debug -> Avvia/Continua;
3. Nella finestra di debug, deseleziona il registro della CPU in Opzioni di debug e vai a Debug -> Finestra di debug -> Deseleziona il registro della CPU.
4. Disabilitare il debug.
5. Quindi reimpostare il punto di interruzione e il debug non eseguirà il flashback;
6. Se si apre nuovamente il registro della CPU, la CPU lampeggerà di nuovo. Andare al passaggio 1.

IL METODO DI CUI SOPRA È APPLICABILE SOLO AL caso in cui il registro della CPU è attivato durante il debug del punto di interruzione. Altri problemi possono anche essere provati usando il metodo sopra.
Non ho idea di cosa si tratti, ma ho trovato questo metodo riavviando accidentalmente Codeblocks ed eseguendo il debug direttamente. Questo significa che il debug del punto di interruzione non può aprire determinate informazioni Windows? Perché?

Registrato

PROBLEMI AD ESEGUIRE L'ESEGUIBILE?

L'eseguibile viene creato in seguito alla compilazione (e al linking) all'interno del progetto nella cartella bin\Debug oppure bin\Release (dipende da cosa è indicato qui 

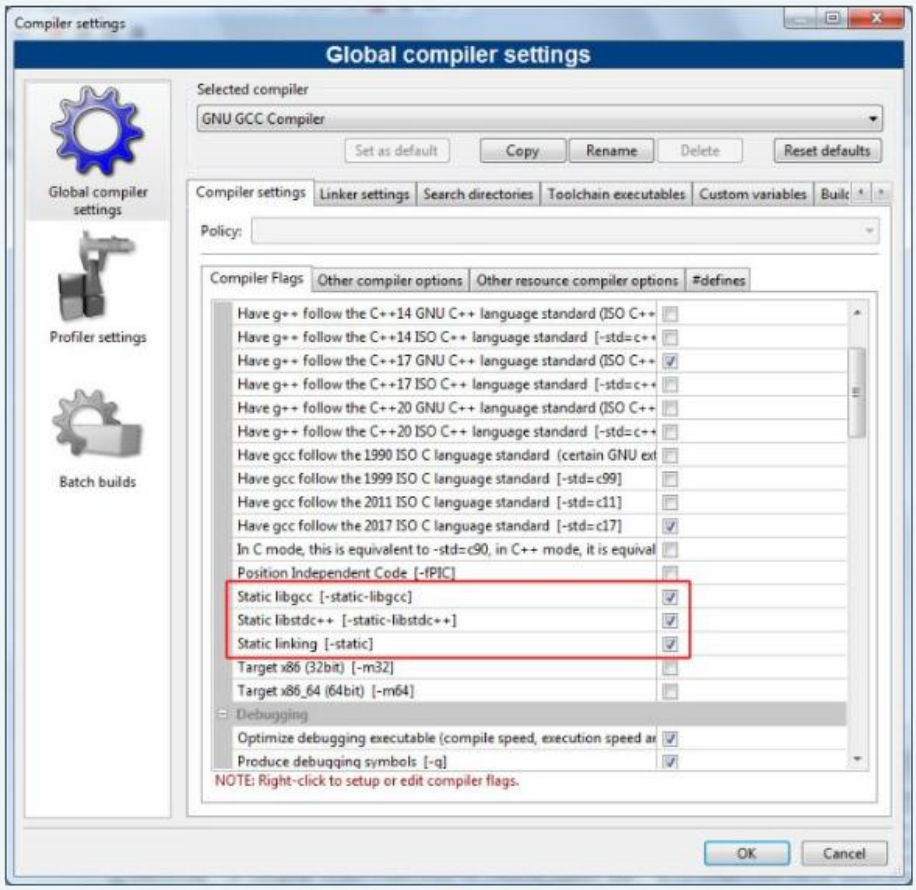
)

Se lo vedi avviarsi e poi terminare subito è perché l'esecuzione è troppo veloce. Aggiungi un getchar(); alla fine per attendere la chiusura.

MANCANO LE DLL E QUINDI L'ESEGUIBILE NON SI AVVIA?

Se il PC non ha installato MinGW questo potrebbe accadere. In tal caso è necessario inserire manualmente le .dll nel PC manualmente oppure aggiungerle all'eseguibile. Quest'ultima soluzione rende l'eseguibile portabile ma aumenta notevolmente il tempo di compilazione, quindi meglio inserire le librerie solo per realizzare la versione di release, come indicato nel post qui sotto:

Code::Blocks, di default, non è impostato per includere le dll nell'exe, ci sono varie soluzioni:
1) è quella che hai adottato, copiare tutte le dll necessarie nella stessa cartella dell'eseguibile.
2) copiare le dll più usate, solitamente 2 o 3, in system32 (sistemi 32bit) o SysWOW64 (sistemi64bit)
3) Includere il percorso di MinGW nelle variabili d'ambiente (Path).
4) Copilare con i flags -static, -static-libgcc, -static-libstdc++
in code::blocks ci sono le apposite spunte...



CompilerSettings.jpg

Se si includono le librerie, l'eseguibile funzionerà anche in sistemi che non hanno MinGW installato e senza dover distribuire insieme all'eseguibile le dll necessarie.
L'unica seccatura che ho riscontrato, è che se si usa il compilatore C, il tempo di compilazione decuplica (improponibile), mentre se si compila in C++ il tempo di compilazione raddoppia (sopportabile).
Attualmente quando sviluppo tengo i flag disabilitati, al termine imposto il flag e l'exe funziona ovunque (ambiente win).

FINESTRA WATCHES NEL DEBUGGER

Come attivare la finestra watches nella perspective di Debug:

