

*Fondamenti di Informatica T1
Ingegneria Elettronica
Ingegneria dell'Automazione
a.a. 2019/2020*

*Lab 00
Introduzione al corso
e a Codelite*

OBIETTIVI

- Conoscere i principi e gli strumenti di base
- Saper esprimere la soluzione a un problema (algoritmo) e codificarlo in un linguaggio di programmazione (C)
- Saper costruire *un singolo componente software* che rispecchi una certa specifica

VALUTAZIONE

La valutazione del corso comprende:

- Una prova di laboratorio (2 ORE e 30)
 - Risoluzione di un problema mediante costruzione di un programma in linguaggio C
 - Occorre dimostrare di saper effettivamente produrre un programma eseguibile
- Una prova (scritta) (1 ORA e 30)
 - Occorre dimostrare di aver assimilato anche le parti “teoriche”

ATTENZIONE !!!!!

- Le due prove sono assolutamente **INDIPENDENTI**, ma entrambe **NECESSARIE** al fine di passare l'esame.
- La prova pratica **DEVE PRECEDERE** lo scritto che può essere effettuato anche in appelli/sessioni d'esami differenti

PREREQUISITI...

- Nessuno...
- ...ma è indispensabile lavorare al calcolatore
per tutta la durata del corso

LABORATORIO

- Attività di esercitazione assistita da tutor
- Attività di esercitazione libera

IL SITO WEB DEL CORSO

<http://lia.disi.unibo.it/Courses/FondT1920-AUT/>

- Il vostro punto di riferimento per
 - materiale didattico (lezioni, esercizi)
 - software gratuito
 - testi degli esami e loro soluzione
- Iscrizioni agli esami ed esiti delle prove

<https://almaesami.unibo.it/>

AMBIENTI DI PROGRAMMAZIONE

- **Linguaggio C**

- **CodeLite** (open source, gratuito, scaricabile dal sito Web) **Strumento di riferimento del corso**
- Possono essere utilizzati altri ambienti (ad es. Turbo C, Dev-C++, Eclipse, ...)
- MS VisualStudio. Ambiente di sviluppo commerciale. Disponibile GRATUITAMENTE in quanto studenti di un corso UNIBO.

ESERCITAZIONI DI LABORATORIO

Andrea Borghesi (andrea.borghesi3@unibo.it)

Giovedì dalle 11.30 alle 14.00 Lab3

–Studenti A-F

Giuseppe Tagliavini (giuseppe.tagliavini@unibo.it)

Giovedì dalle 12.00 alle 14.30 Lab4

–Studenti

REGOLE D'ESAME

- Voto = media pesata (2/3 pratica, 1/3 scritto)
- È necessaria la sufficienza alla prova pratica per accedere allo scritto
- I voti restano validi un anno
- In caso di consegna il voto precedente viene perduto
- È possibile sostenere le prove in appelli differenti
- È necessaria la sufficienza in entrambe le prove per superare l'esame
- Il voto (pubblicato) dello scritto è effettivamente la media pesata

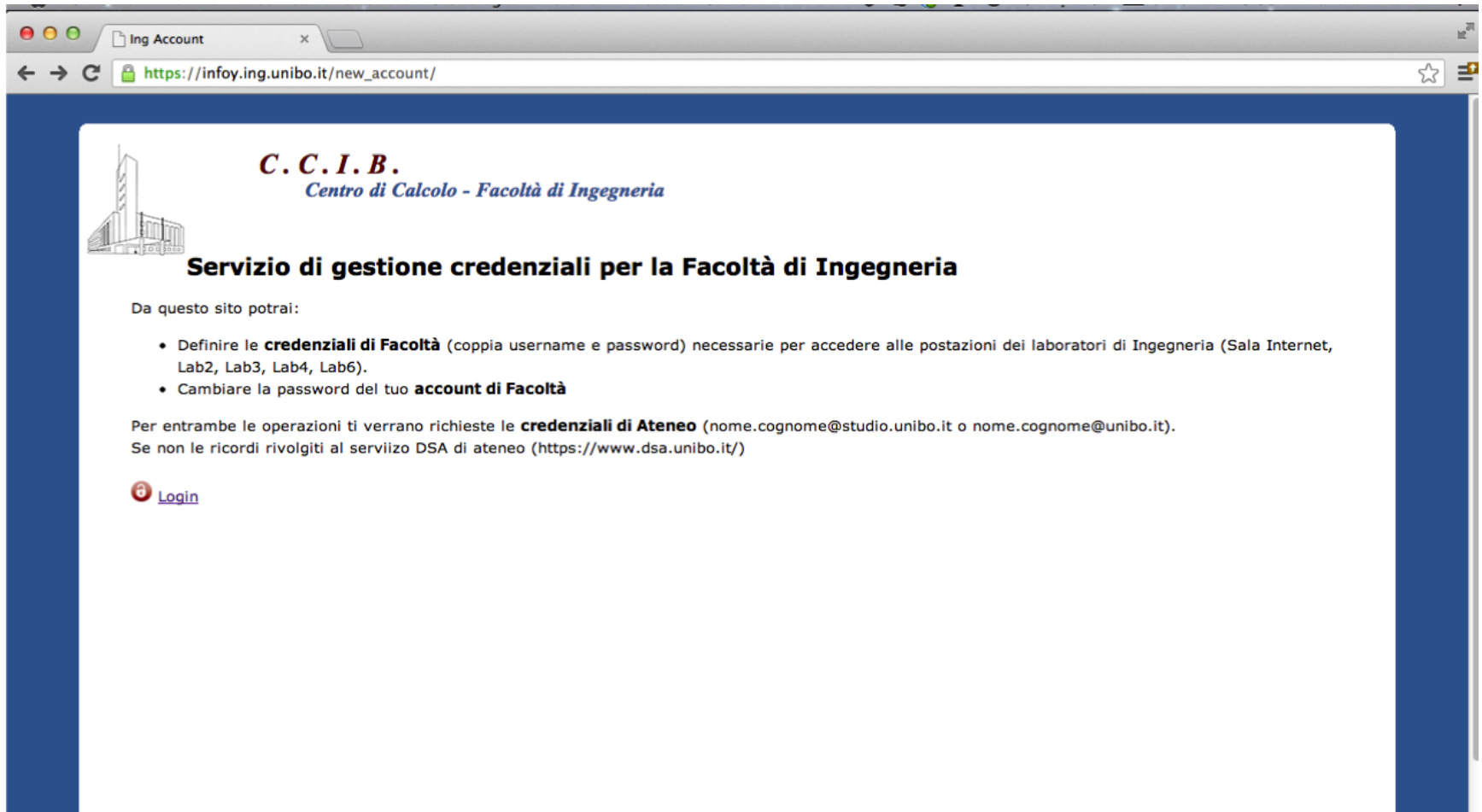
REGOLE D'ESAME

- Ci sono molte soluzioni possibili
- Guardate gli esercizi risolti solo quando vi bloccate
- Abituativi a scrivere codice che esegua
- Abituativi a scrivere codice senza errori
- Consegnate solo codice eseguibile alla prova pratica di laboratorio

DATE D'ESAME

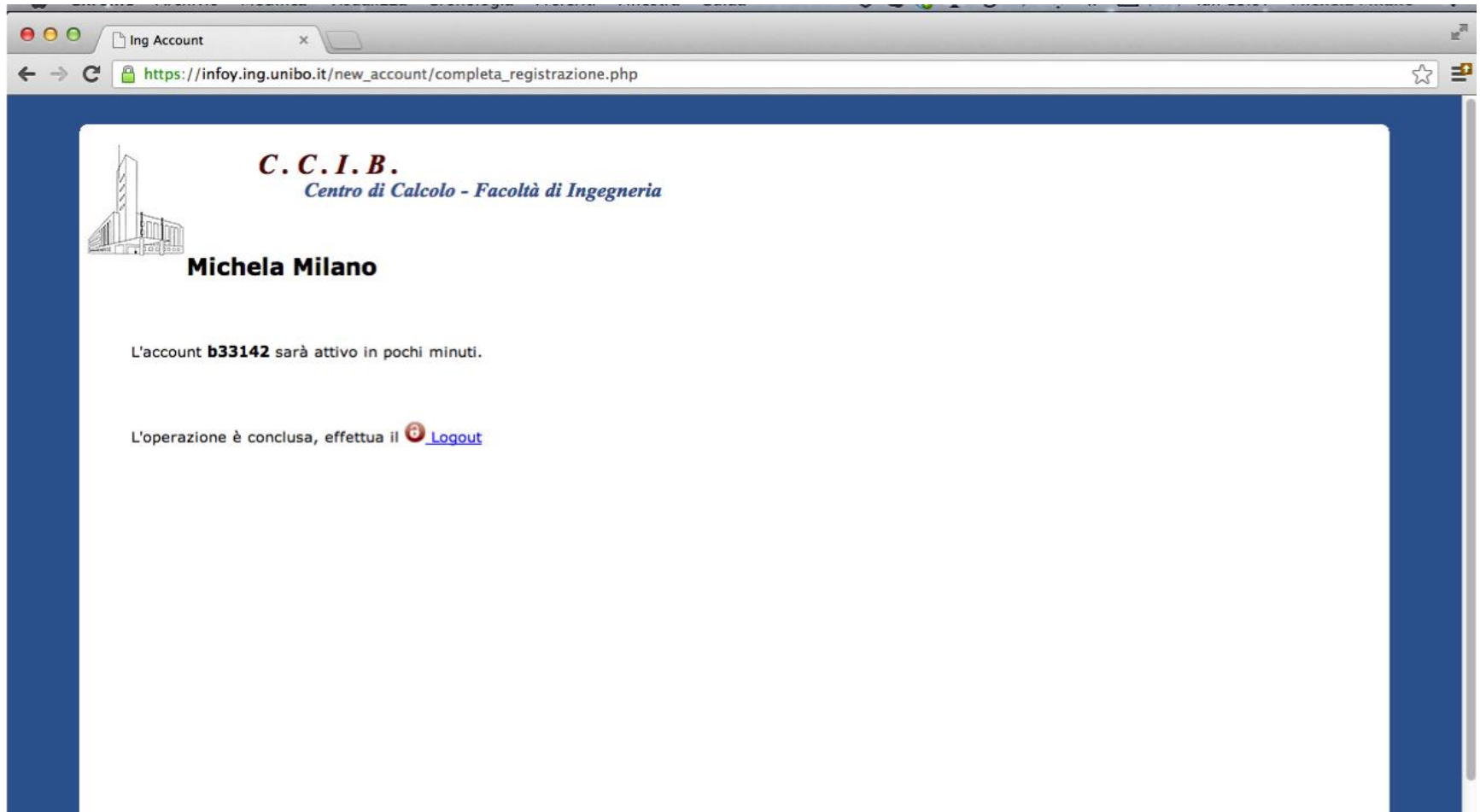
- Pratica: 15 Giugno, 11:00-12:30, presentarsi in Lab4
- Scritto: 19 Giugno, 10:00-11:30, Aule 6.1 e 6.2
- Pratica: 6 Luglio, 11:00-12:30, presentarsi in Lab4
- Scritto: 10 Luglio, 10:00-11:30, Aule 6.1 e 6.2
- Pratica: 8 Settembre, 11:00-12:30, presentarsi in Lab4
- Scritto: 11 Settembre, 10:00-11:30, Aule 2.7B e 2.4

Creazione Account... (2)



The screenshot shows a web browser window with the address bar displaying https://infoy.ing.unibo.it/new_account/. The page has a blue header and a white main content area. On the left, there is a small icon of a building. To its right, the text **C.C.I.B.** is displayed in a large, bold, serif font, followed by *Centro di Calcolo - Facoltà di Ingegneria* in a smaller, italicized serif font. Below this, the heading **Servizio di gestione credenziali per la Facoltà di Ingegneria** is

Creazione Account... (5)



Costruzione di un'Applicazione

Per costruire un'applicazione occorre:

- **compilare il file (o *i* file se più d'uno) che contengono il testo del programma (file *sorgente*)**

Il risultato sono uno o più file *oggetto*.

- **collegare i file oggetto l'uno con l'altro e con le librerie di sistema.**

Preprocessore

Il **preprocessore** agisce prima del **compilatore** manipolando il testo del file sorgente:

- Include porzioni di testo prese da altri file (direttiva *#include*)
- Effettua sostituzioni di testo (direttiva *#define*)
- Elimina parti di testo (ad esempio i commenti)

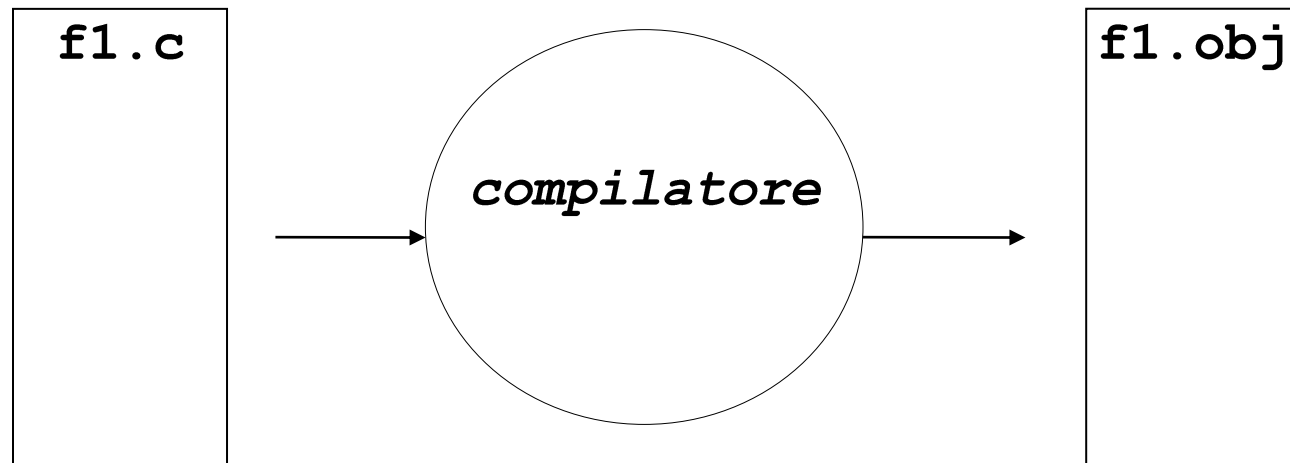
Il preprocessore non è un compilatore

Non è in grado di rilevare eventuali errori di sintassi

Compilazione di un'Applicazione

1) Compilare il file (o i file se più d'uno) che contengono il testo del programma

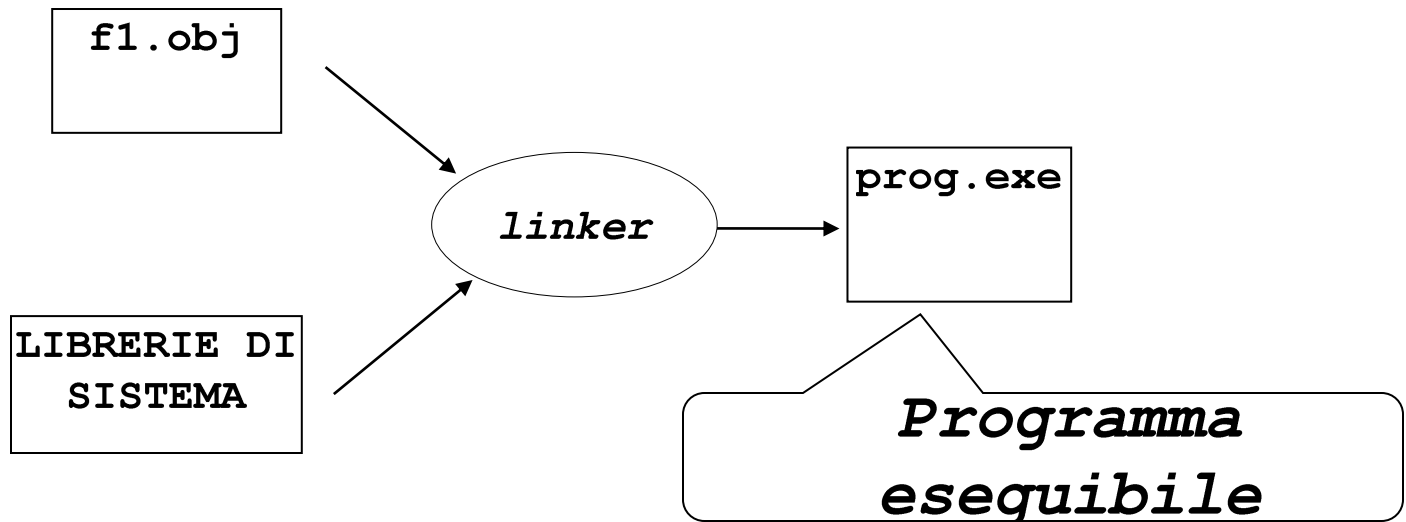
- File *sorgente*: estensione **.c**
- File *oggetto*: estensione **.o** o **.obj**



Collegamento (Linking) di un'Applicazione

2) Collegare il file (o *i* file) oggetto fra loro e con le librerie di sistema

- File oggetto: estensione **.o** o **.obj**
- File eseguibile: estensione **.exe** o nessuna



Collegamento (Linking) di un'Applicazione

LIBRERIE DI SISTEMA:

insieme di componenti software che consentono di interfacciarsi col sistema operativo, usare le risorse da esso gestite, e realizzare alcune "istruzioni complesse" del linguaggio

Ambienti Integrati

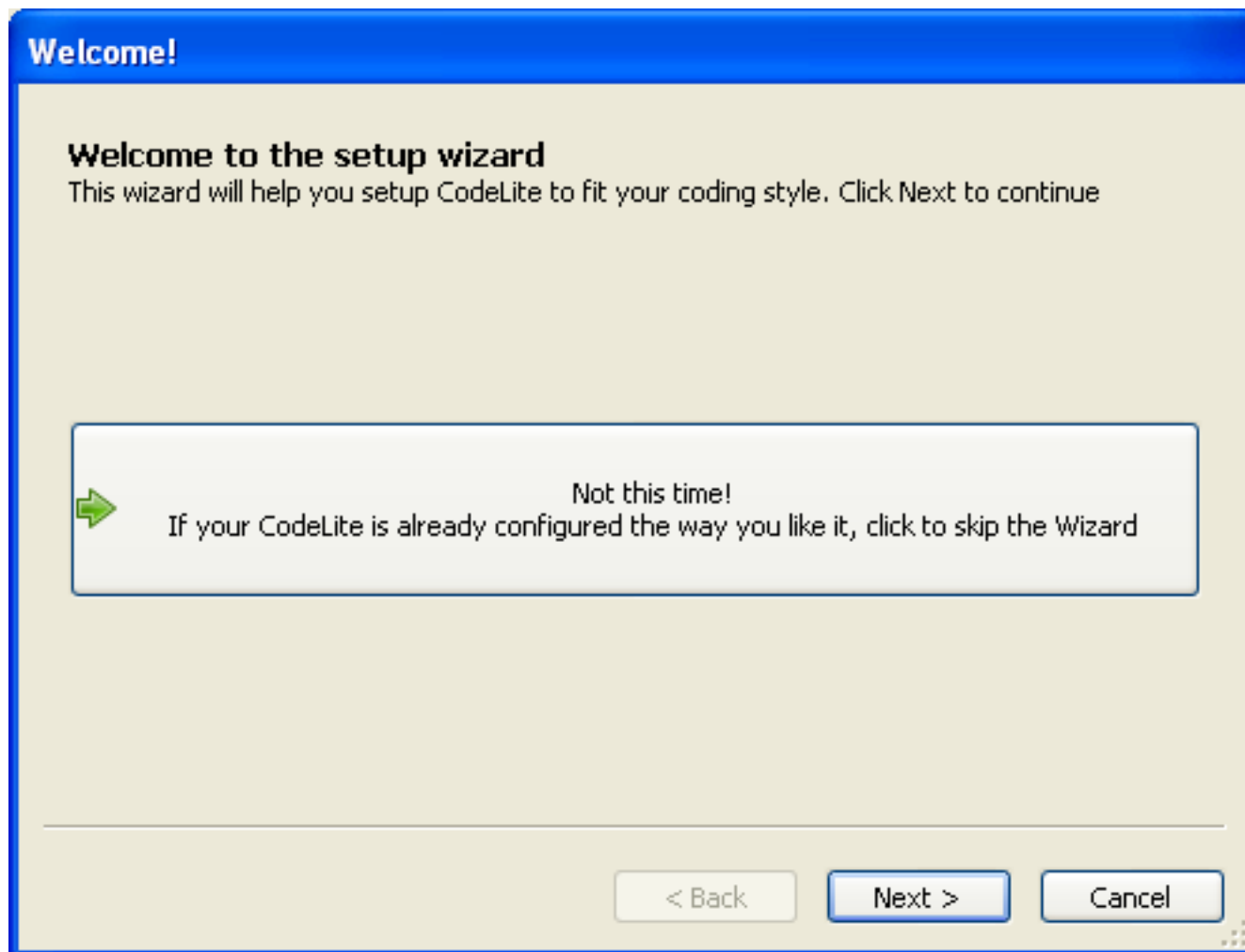
Oggi, gli *ambienti di lavoro integrati* automatizzano la procedura:

- **compilano i file sorgente (se e *quando necessario*)**
- **invocano il linker per costruire l'eseguibile**

ma per farlo devono sapere:

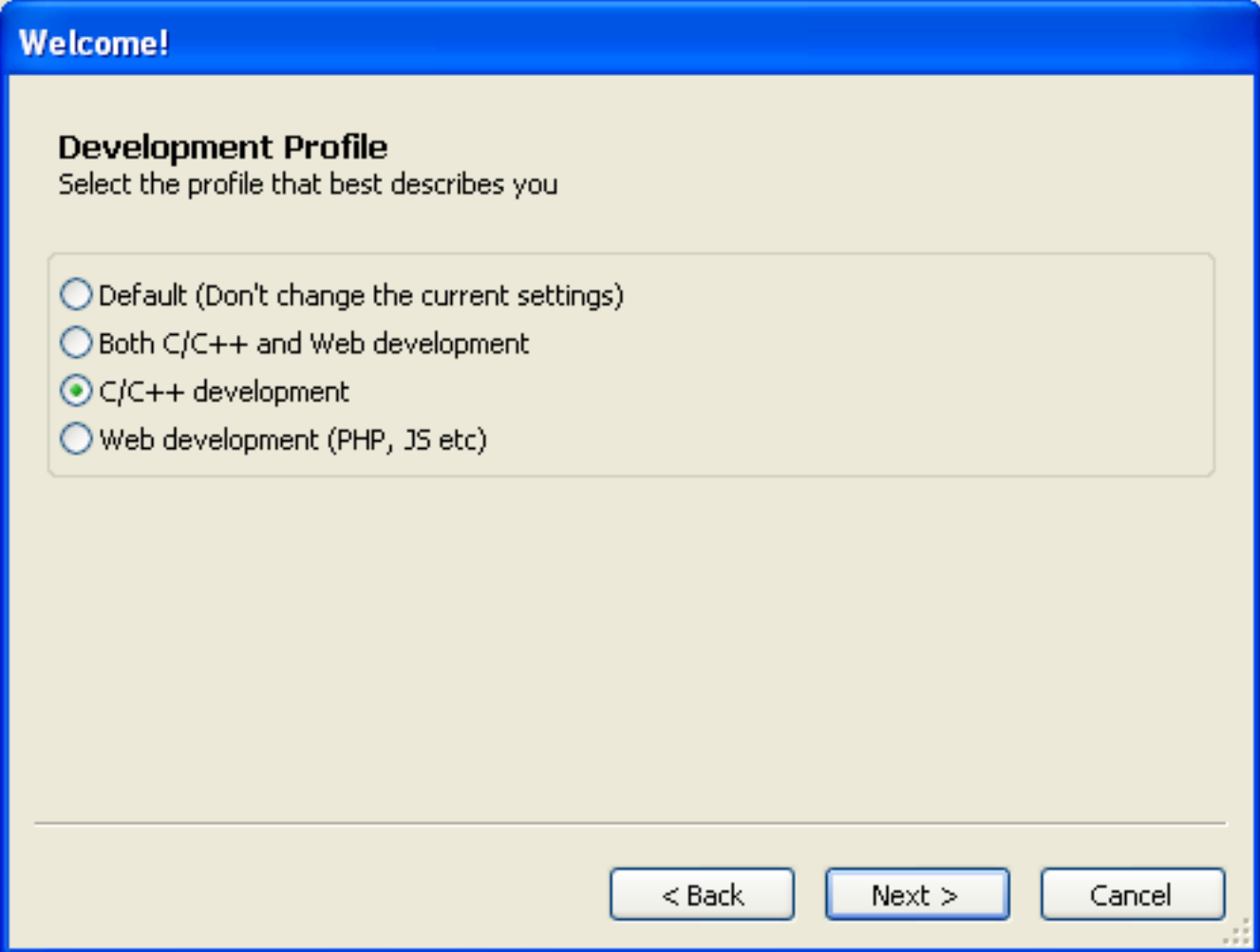
- ***quali file sorgente costituiscono l'applicazione***
- ***il nome dell'eseguibile da produrre.***

Prima Configurazione (2)



Prima Configurazione (3)

Selezionare C/C++ come Development Profile



The image shows a Windows-style dialog box titled "Welcome!". Inside, there is a section titled "Development Profile" with the instruction "Select the profile that best describes you". Below this, there are four radio button options: "Default (Don't change the current settings)", "Both C/C++ and Web development", "C/C++ development" (which is selected, indicated by a green dot), and "Web development (PHP, JS etc)". At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel".

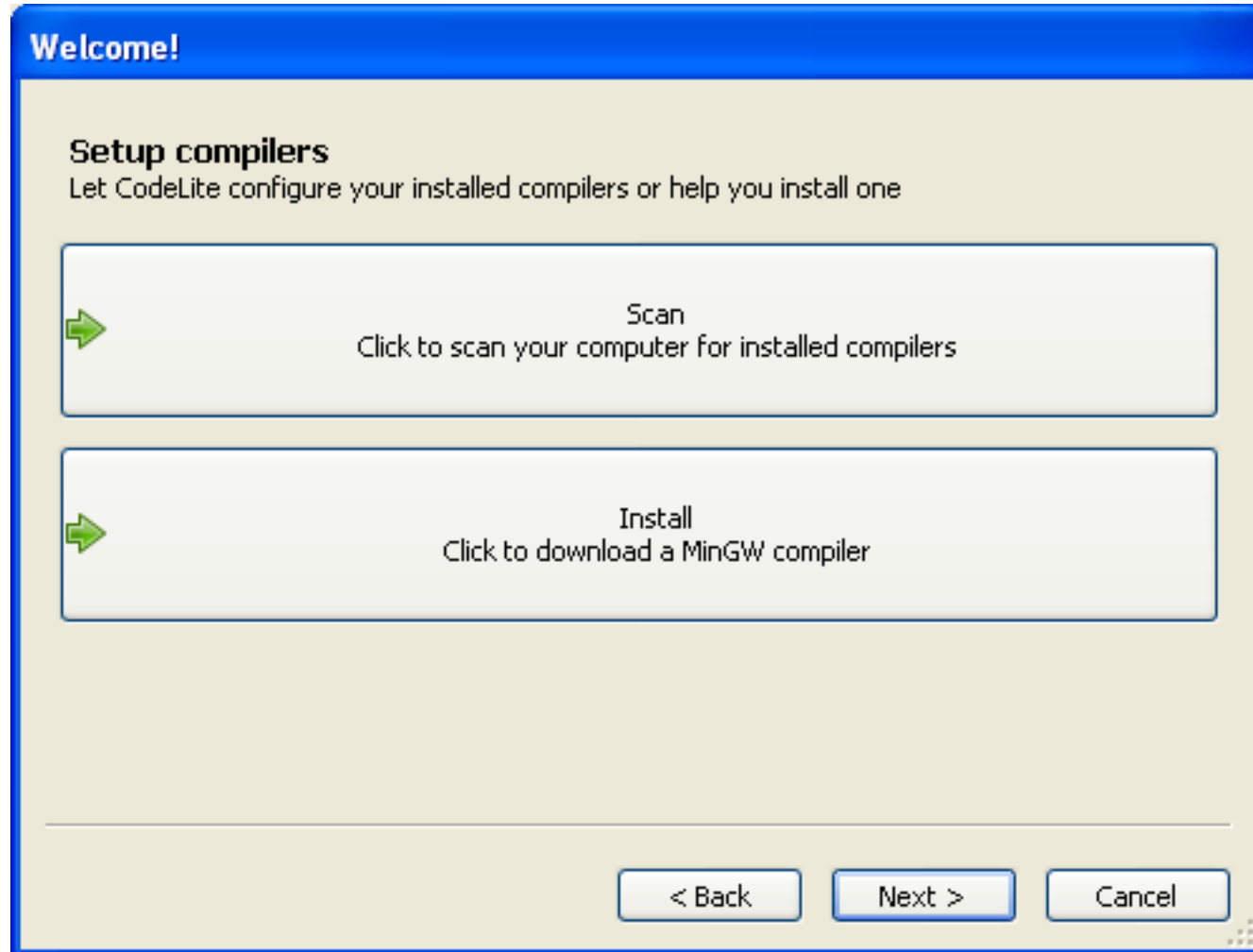
Welcome!

Development Profile
Select the profile that best describes you

- ☐ Default (

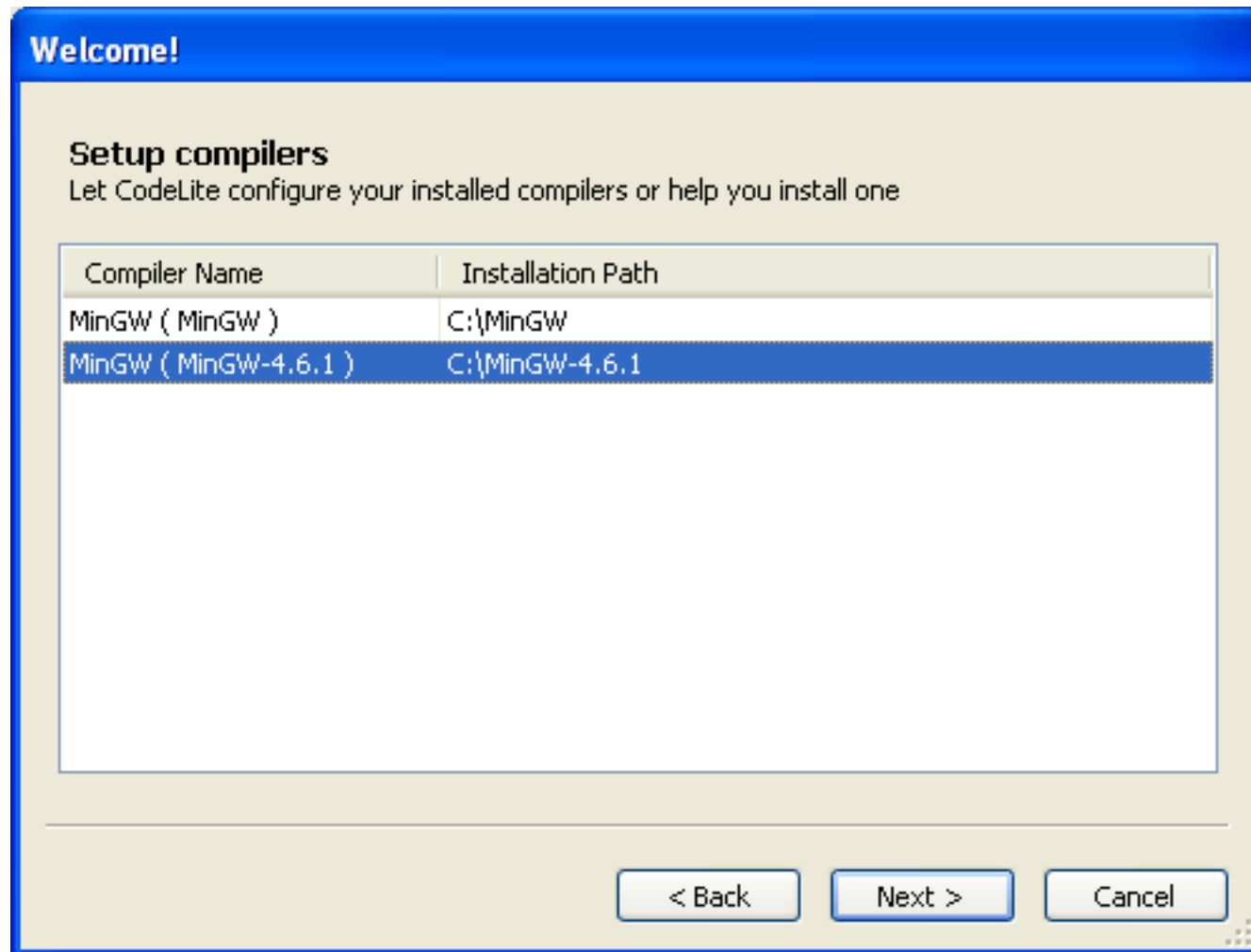
Prima Configurazione (4)

Selezionare **Scan** per rilevare i compilatori installati



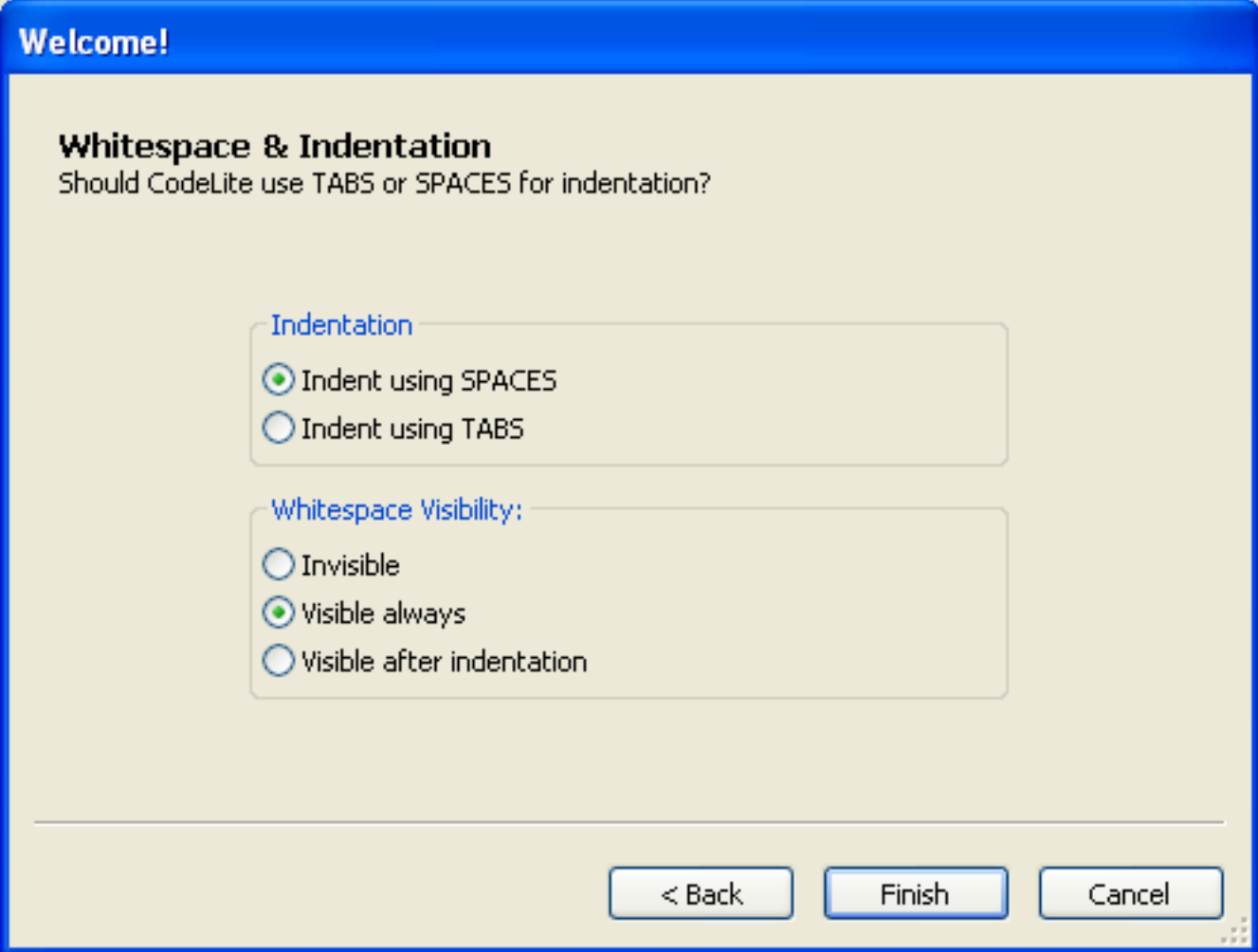
Prima Configurazione (5)

Selezionare **MinGW** come compilatore



Prima Configurazione (7)

Selezionare il tipo di indentazione



Welcome!

Whitespace & Indentation
Should CodeLite use TABS or SPACES for indentation?

Indentation

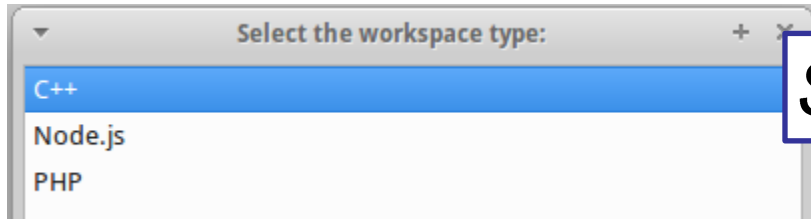
- ☒ Indent using SPACES
- ☐ Indent using TABS

Whitespace Visibility:

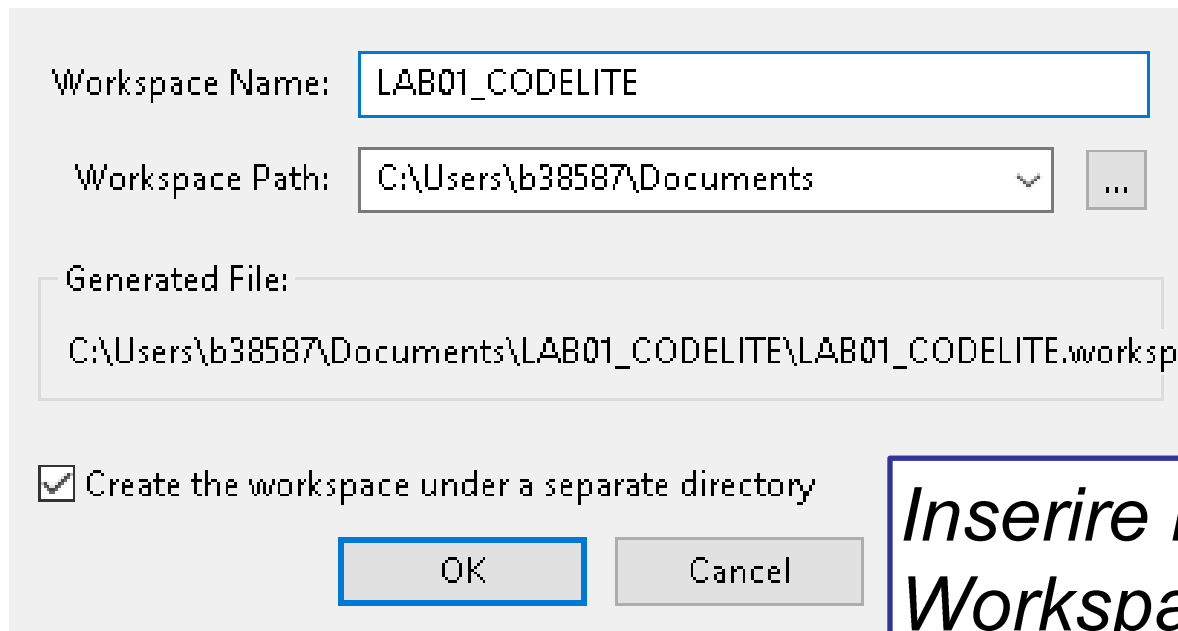
- ☐ Invisible
- ☒ Visible always
- ☐ Visible after indentation

< Back

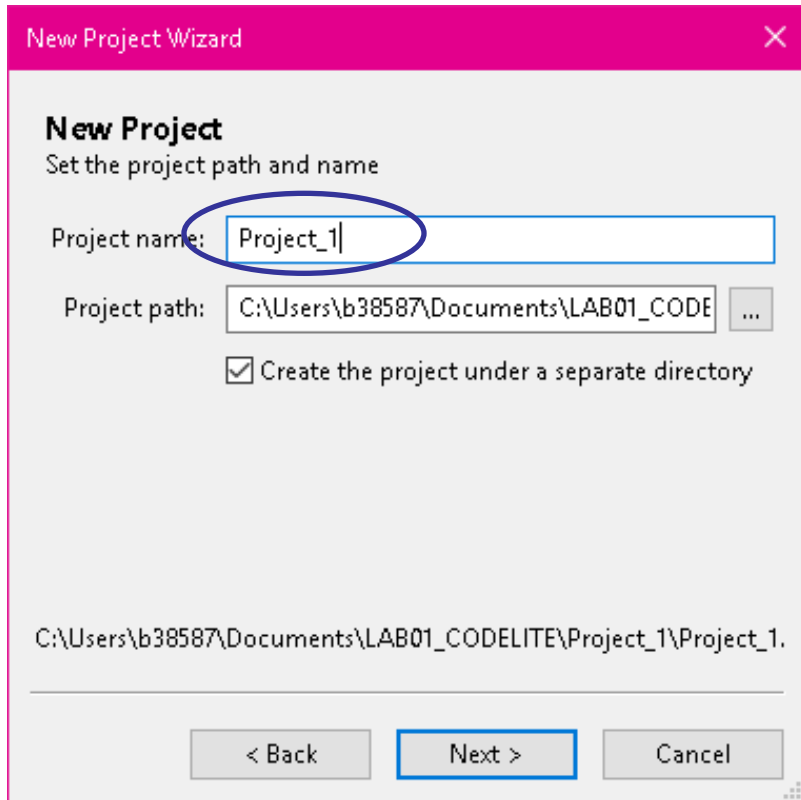
Progetti in CodeLite



Selezionare C++



Progetti in Codelite



New Project Wizard

New Project
Set the project path and name

Project name:

Project path: ...

☒ Create the project under a separate directory

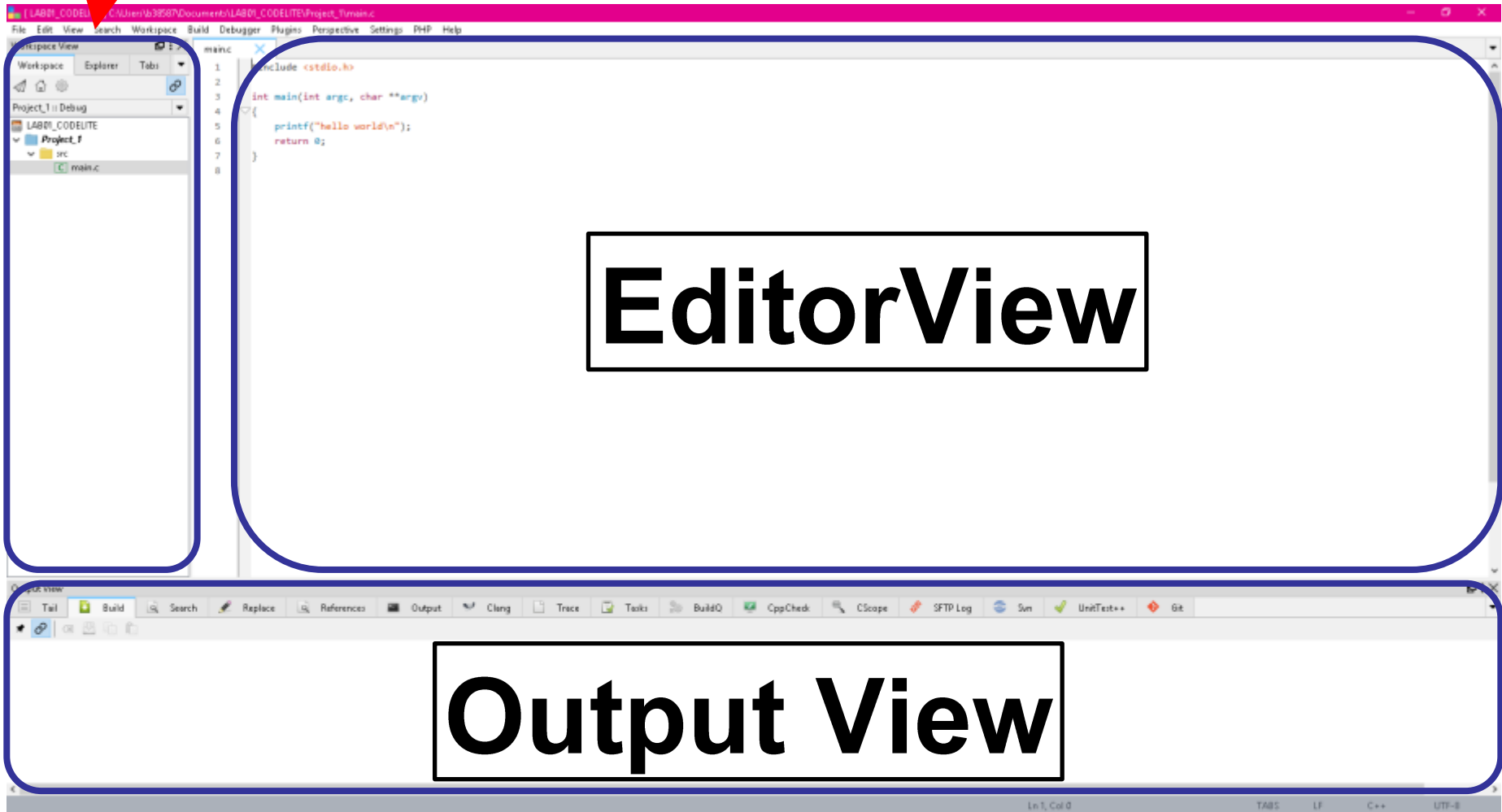
C:\Users\b38587\Documents\LAB01_CODELITE\Project_1\Project_1.

< Back **Next >** Cancel

Inserire il nome del progetto

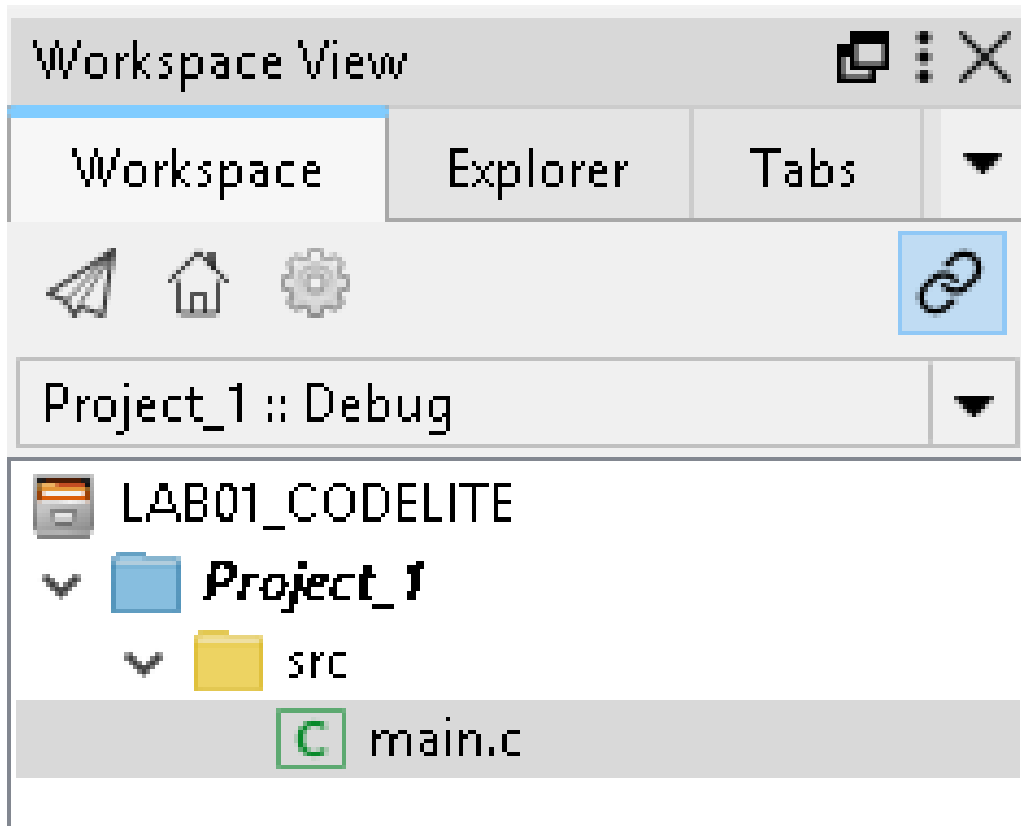
Progetti in Codelite

Workspace View



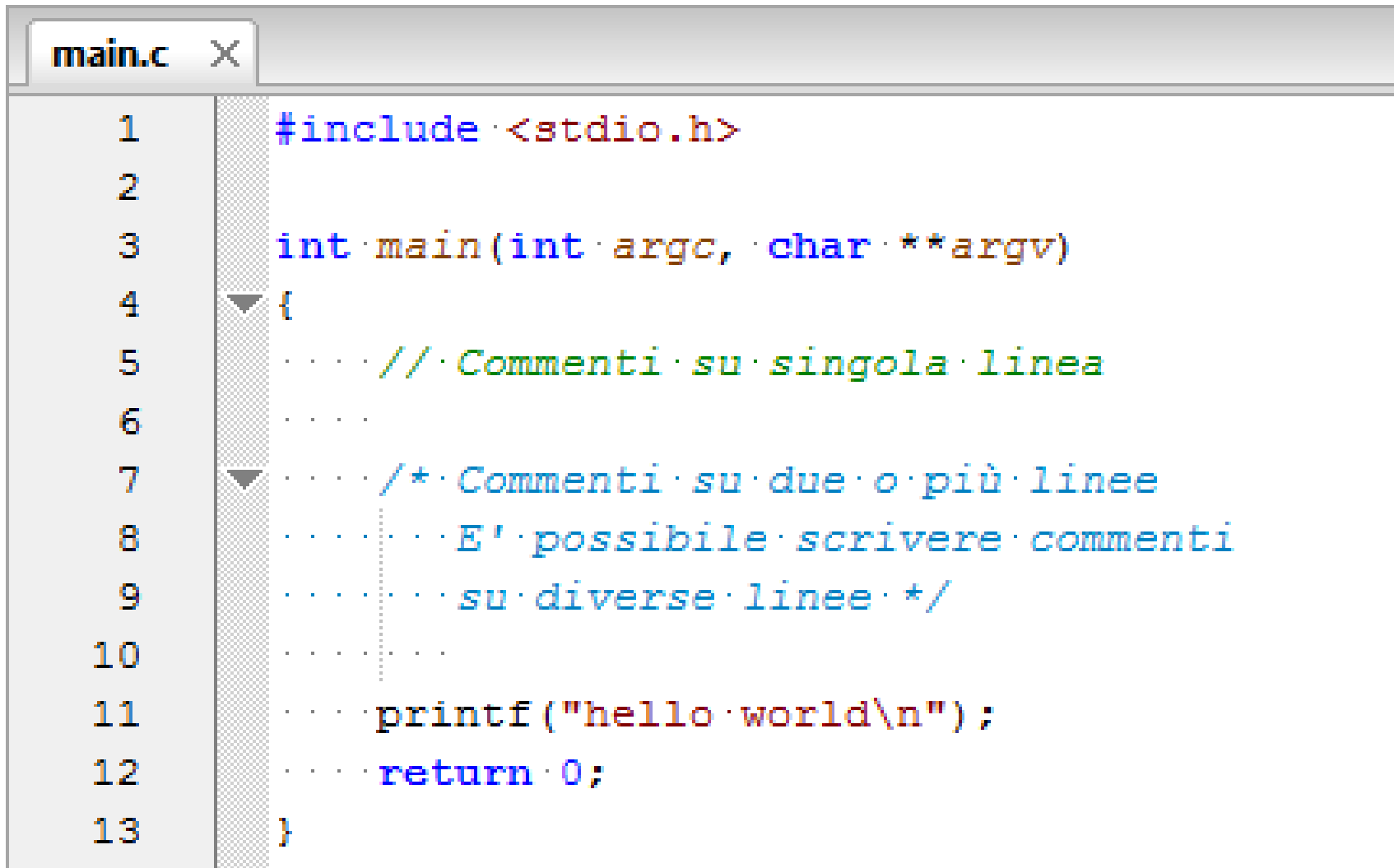
Progetti in Codelite

Workspace View



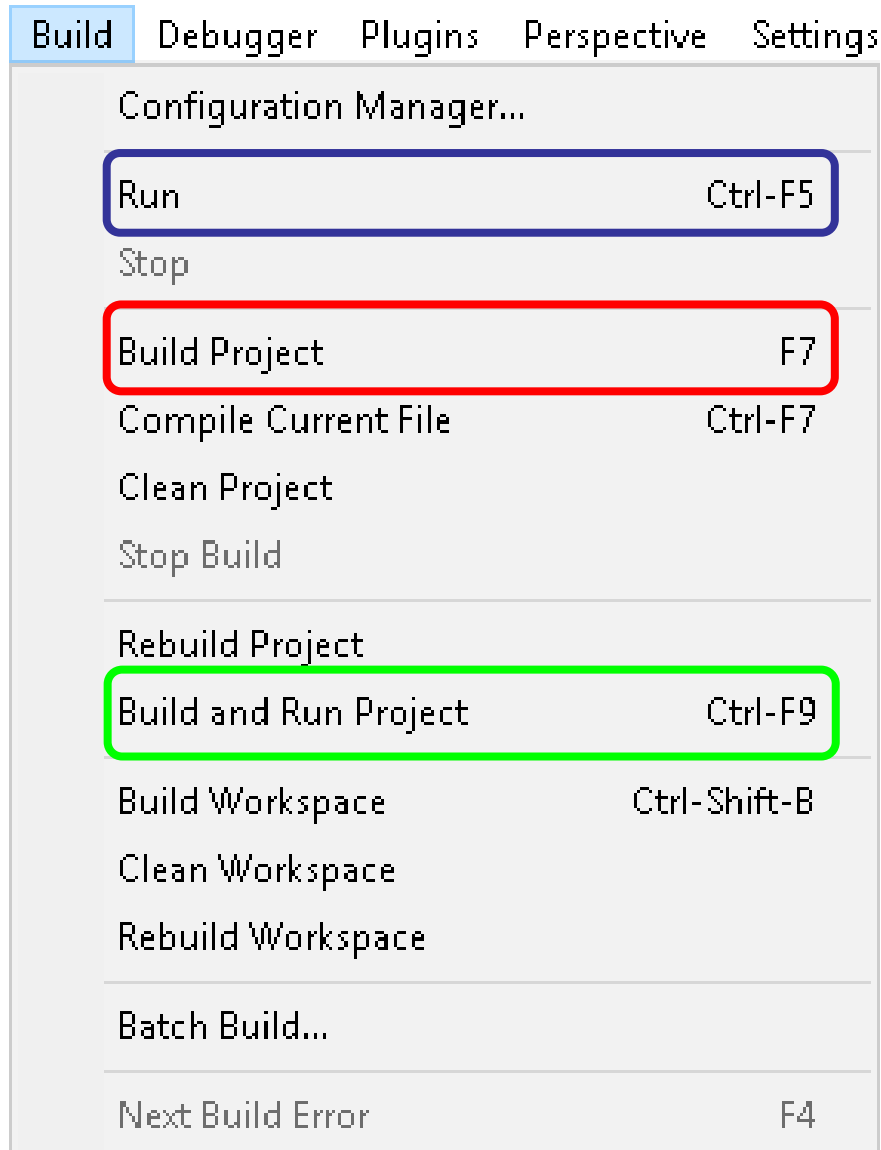
Progetti in CodeLite

EditorView



```
1  #include <stdio.h>
2
3  int main(int argc, char **argv)
4  {
5      ... // Commenti su singola linea
6      ...
7      ... /* Commenti su due o più linee
8           ... E' possibile scrivere commenti
9           ... su diverse linee */
10     ...
11     ... printf("hello world\n");
12     ... return 0;
13 }</
```

Progetti in Codelite



Progetti in Codelite

QUANDO IN UN **WORKSPACE** SONO PRESENTI
DIVERSI **PROGETTI**, IL **PROGETTO ATTIVO** È
SCRITTO IN GRASSETTO

 LAB01_CODELITE

>  *Project_1*

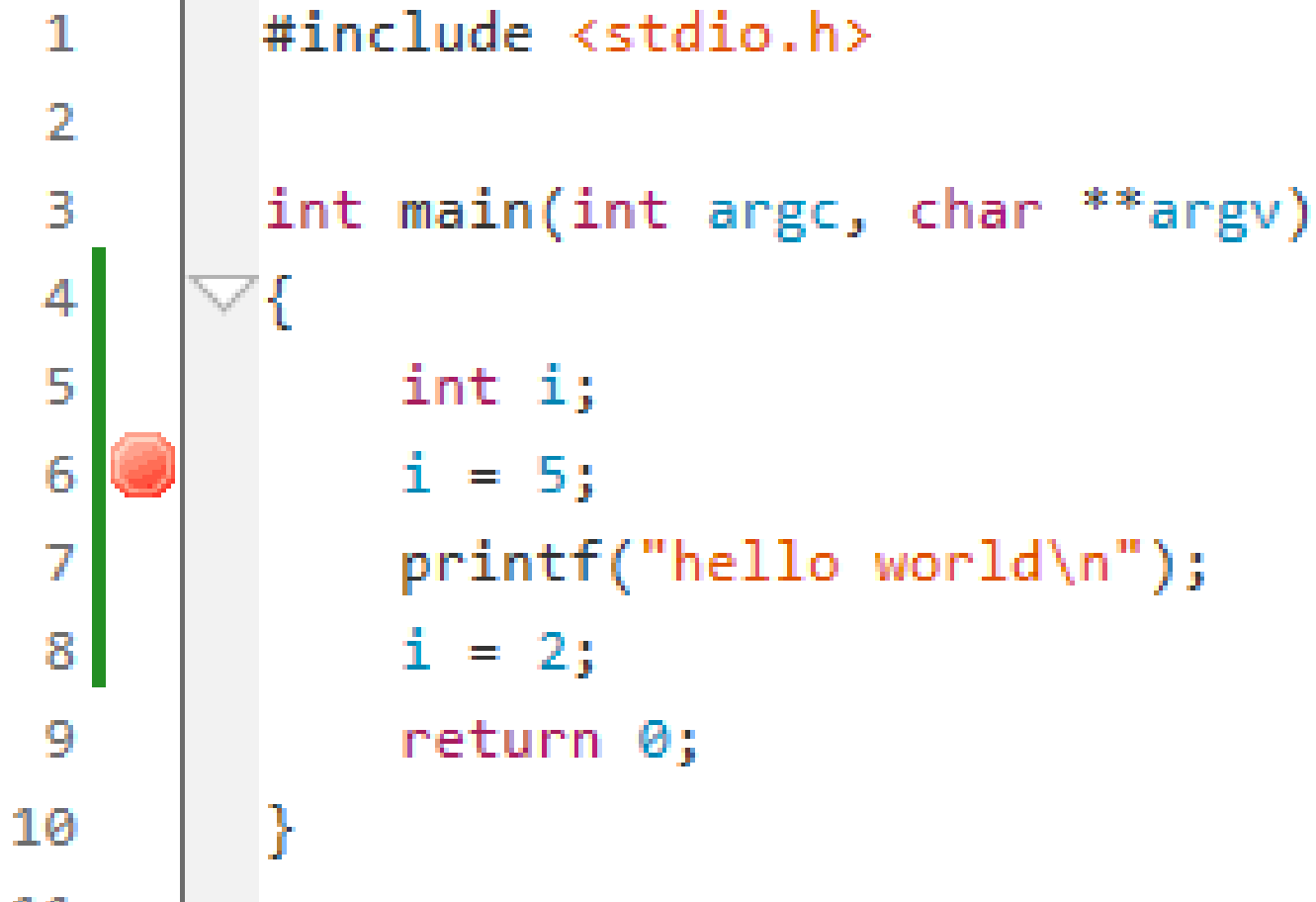
>

ESAMIX

<http://esamix.labx>

Debugger: Locals

```
1      #include <stdio.h>
2
3      int main(int argc, char **argv)
4      {
5          int i;
6          i = 5;
7          printf("hello world\n");
8          i = 2;
9          return 0;
10     }
```

A screenshot of a code editor with a light gray background. The code is a C program for a debugger. It has 10 lines of code. Line 1: #include <stdio.h>. Line 2: (empty). Line 3: int main(int argc, char **argv). Line 4: {. Line 5: int i;.

