



**POLITECNICO**  
MILANO 1863

# **INFORMATICA A – Prof. Fuggetta**

**Laboratorio - Sessione introduttiva**  
prof. Davide Mazza, Ph.D. MBA

# Cos'è e a cosa serve il laboratorio di Informatica A

**Tre incontri con frequenza obbligatoria.**

**Squadra 1** - 23/10, 6/11, 27/11

**Squadra 2** - 30/10, 13/11, 4/12

**Squadra 3** - 23/10, 6/11, 27/11

**No esame finale.**

**Contatti:**

[davide.mazza@polimi.it](mailto:davide.mazza@polimi.it)

# Cosa serve per programmare

## L'IDE: Integrated Development Environment

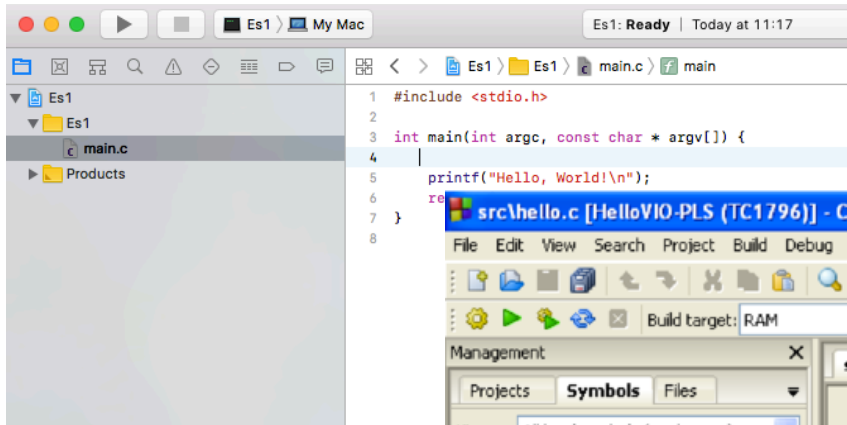
L'IDE, detto anche ambiente di sviluppo, permette di:

- Scrivere il codice
- Verificare eventuali errori di sintassi
- Compilare il codice
- Creare e avviare il file eseguibile

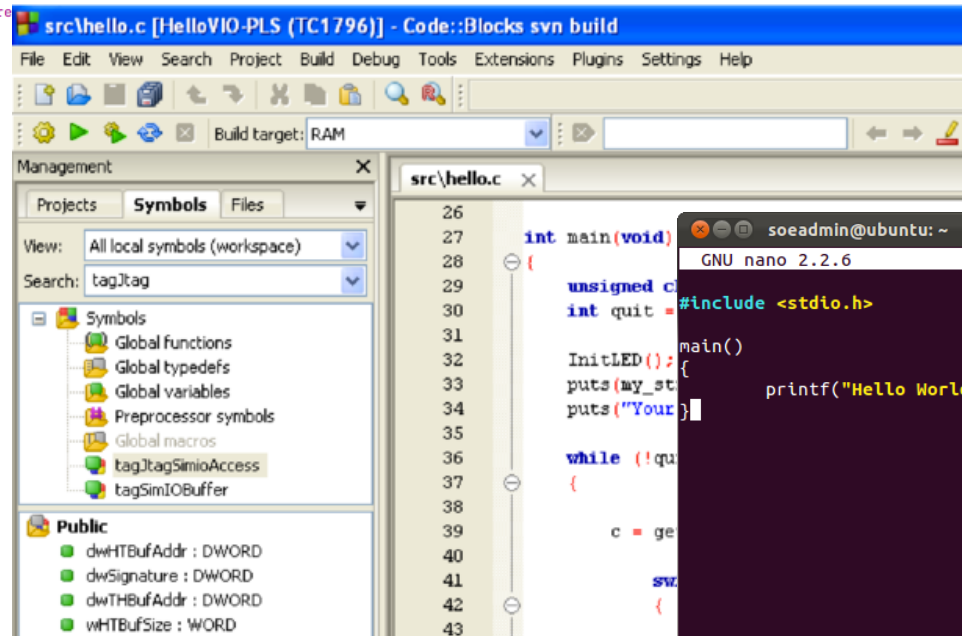
Il codice viene scritto in un file di testo con l'estensione “.c”.

# Cosa serve per programmare

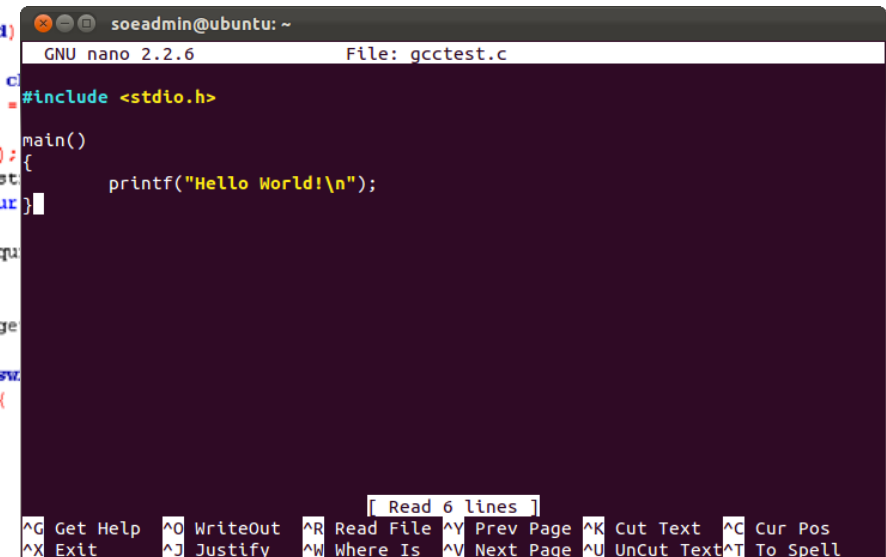
## L'IDE: Integrated Development Environment



Xcode x Mac



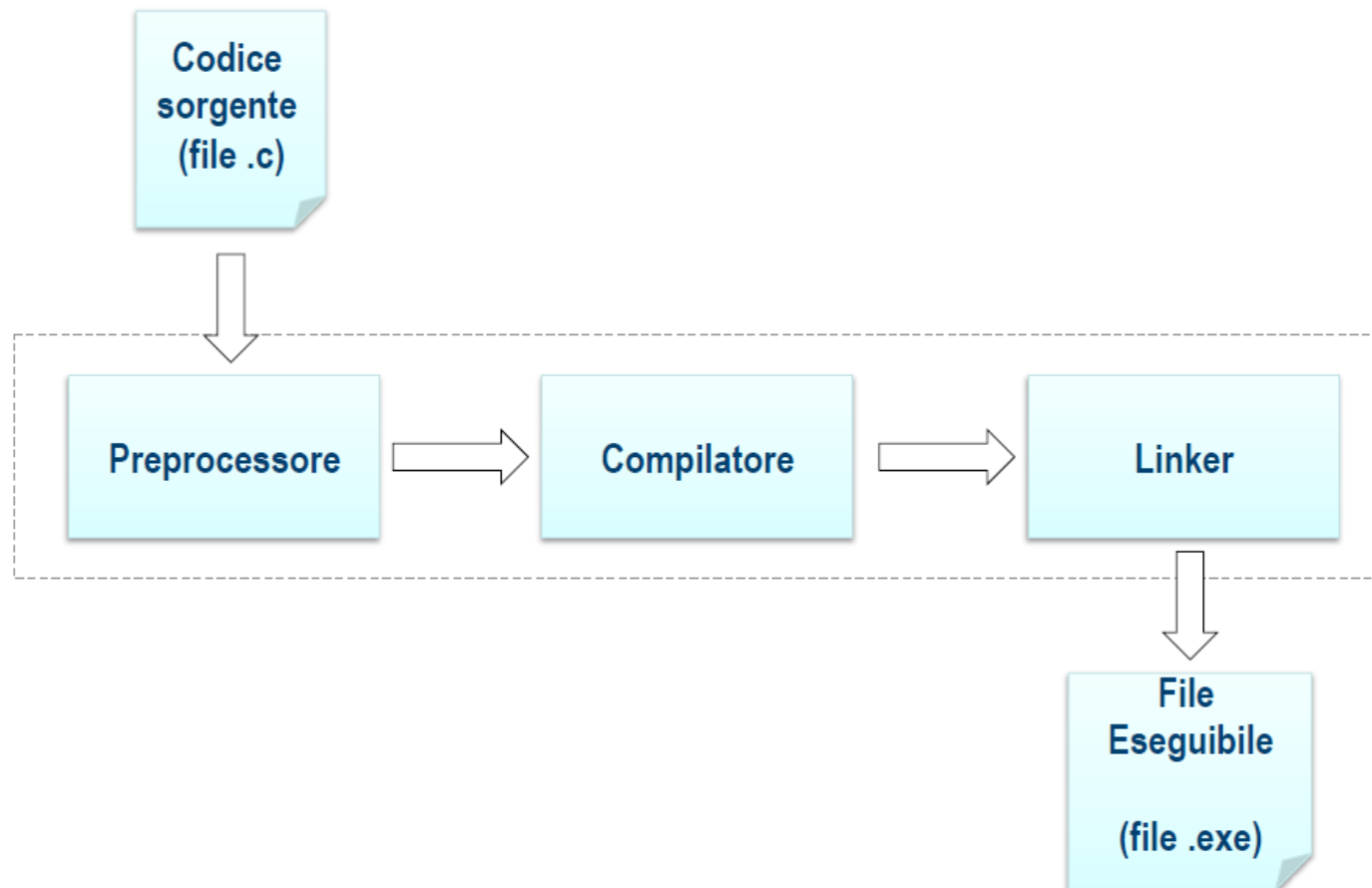
Code:blocks



Unix command line

# Cosa serve per programmare

## L'IDE: Integrated Development Environment

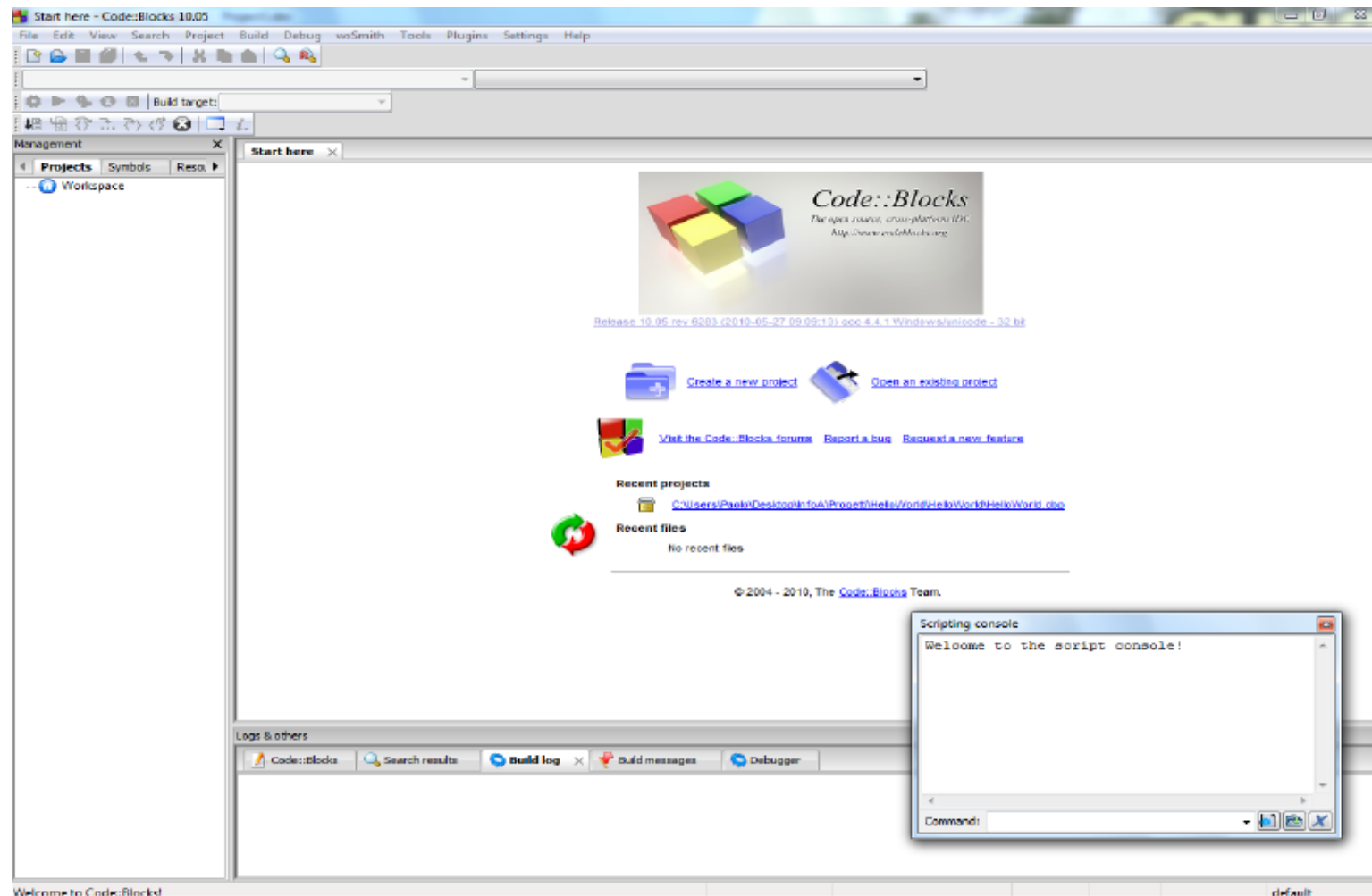


Durante le esercitazioni di laboratorio useremo **Code:Blocks**

Alternative:

- Xcode (solo su MacOS)
- Dev-Cpp (Win Os)
- Visual Studio Code (all)
- [www.repl.it](http://www.repl.it) (da browser, anche su tablet)

# Code::Blocks sui PC del laboratorio



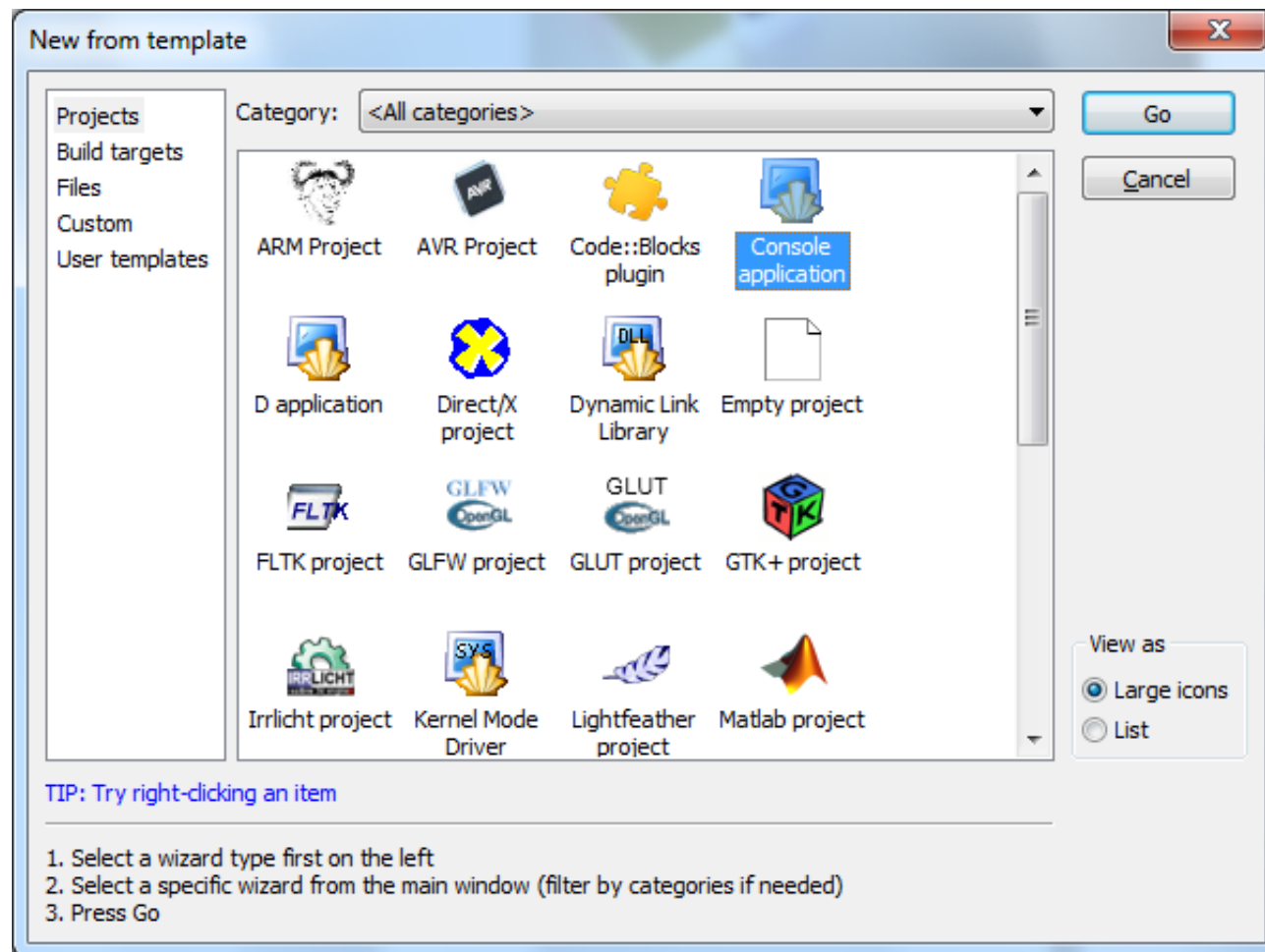


# Code:Blocks

## Creare un nuovo progetto "C"

Per creare un nuovo progetto:

1. Menu file
2. New
3. Project
4. Select **Console application**

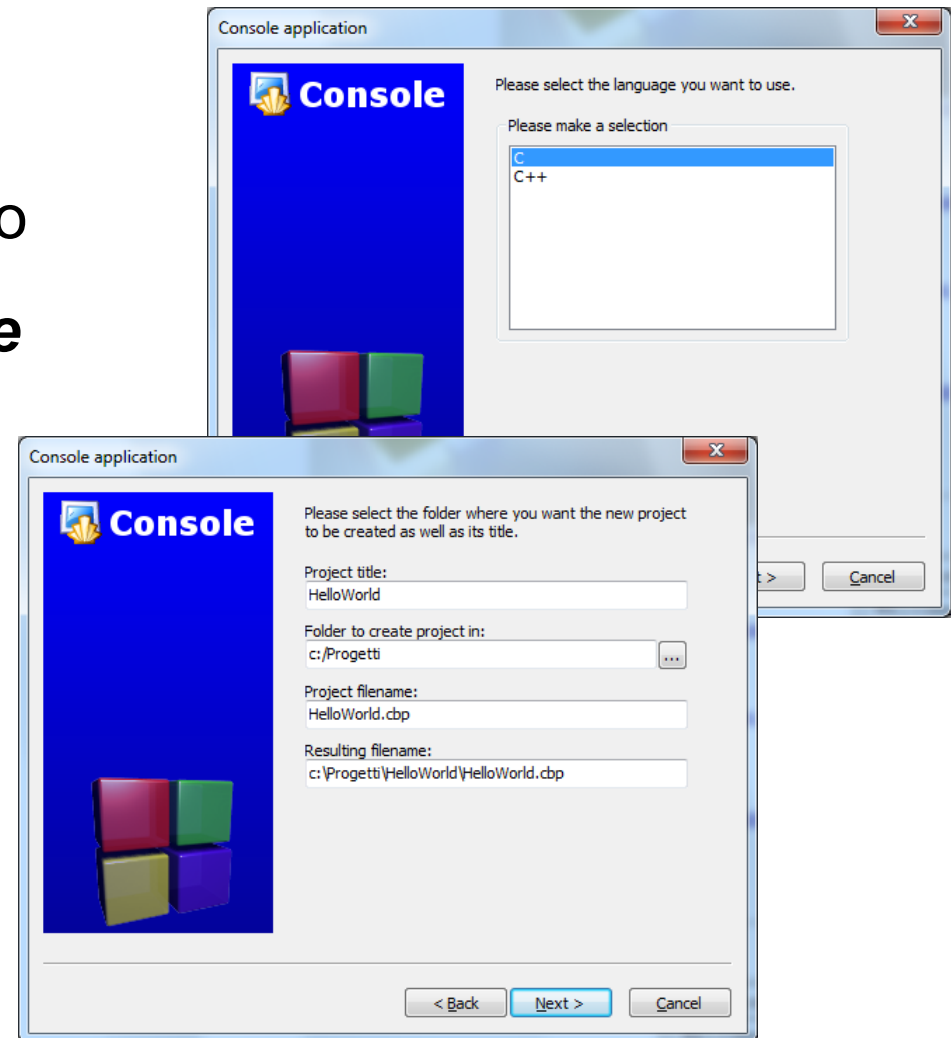




# Code:Blocks

## Creare un nuovo progetto “C”

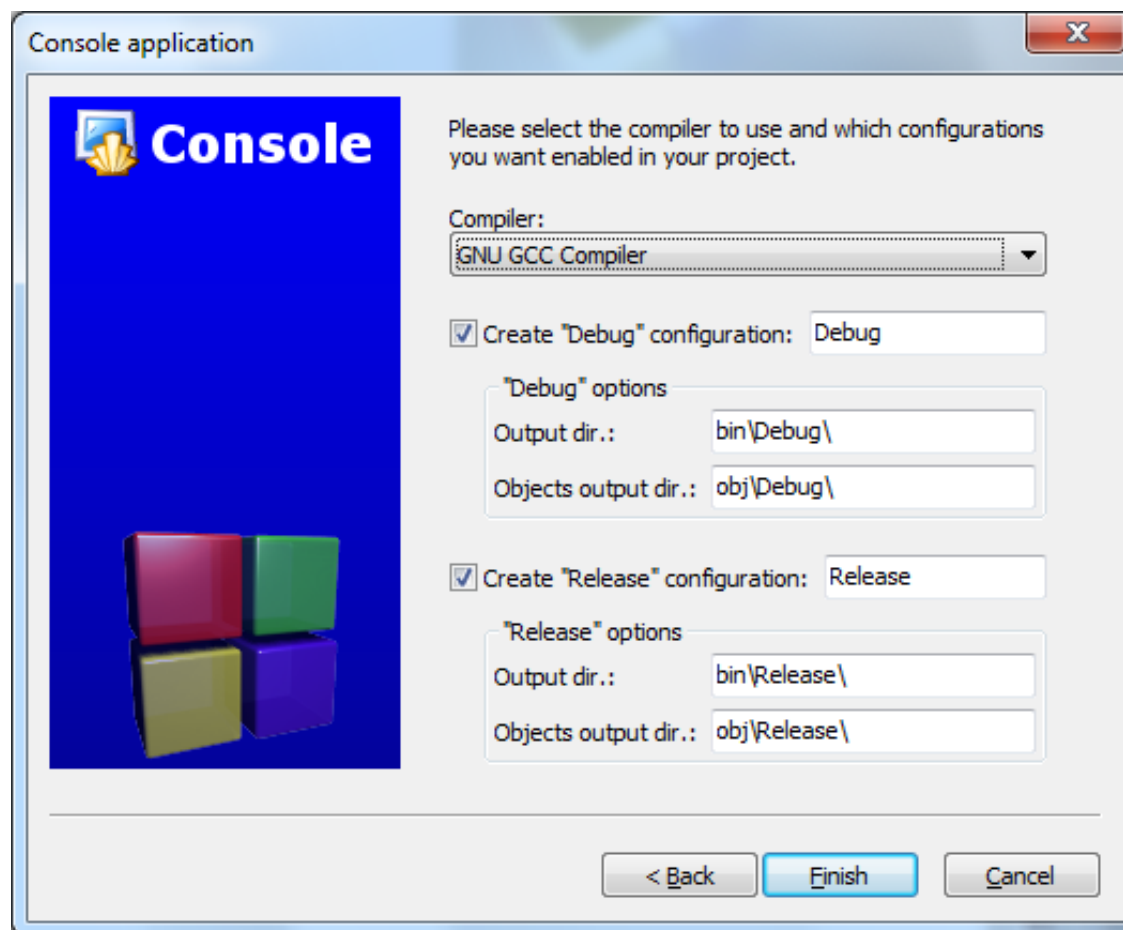
5. Selezionare il linguaggio C
6. Selezionare il nome del progetto ed il percorso. ***Il percorso deve essere la cartella: ...Propri nome/Documents/***



# Code:Blocks

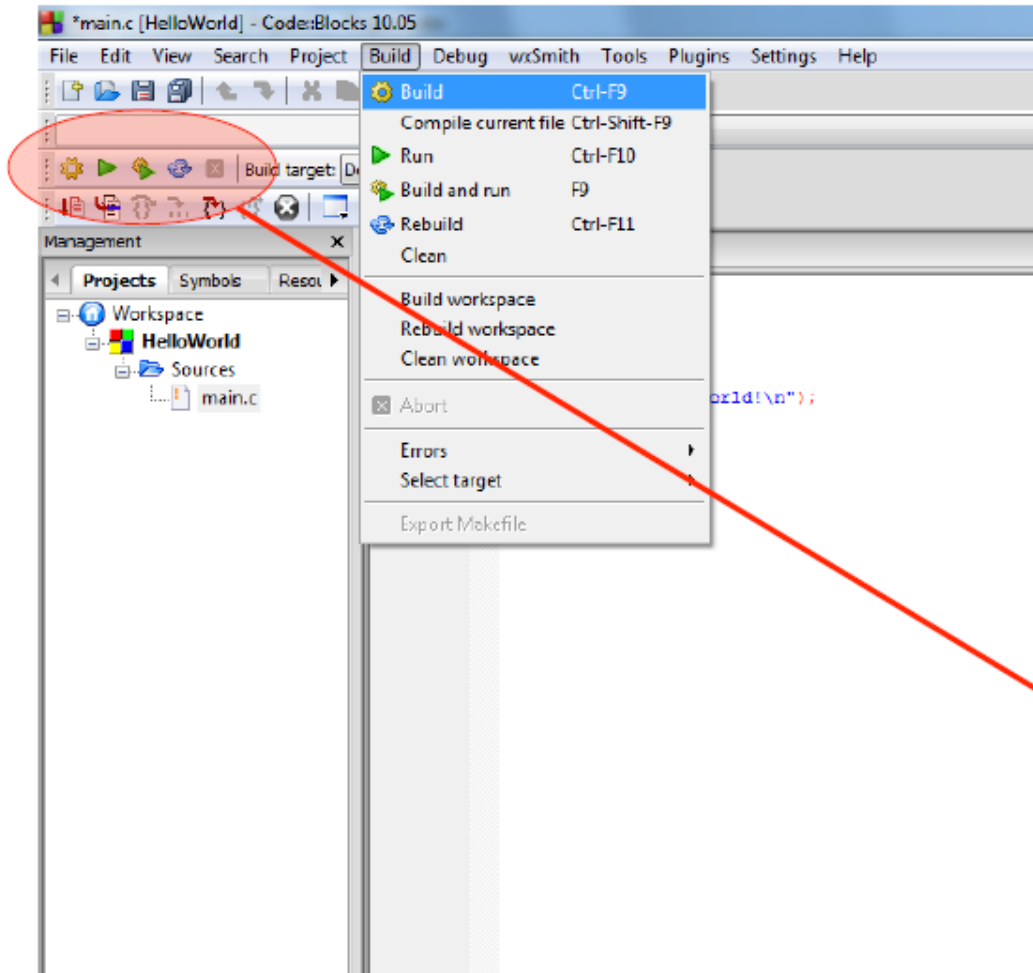
## Creare un nuovo progetto "C"

### 7. Scegliere il compilatore **GNU GCC Compiler**



# Code:Blocks

## Creare un nuovo progetto "C"



Per compilare ed eseguire il programma si può utilizzare il menu Build o le icone.

**Prima di compilare, è necessario salvare i file modificati**

**Build > Compila**

**Run > Esegue**

# Code:Blocks

## Primo programma

Proviamo a scrivere il primo programma che stampa a video il proprio nome e cognome.

```
#include <stdio.h>

int main ()
{
    printf("Il mio nome è ...");
    return 0;
}
```

# Code:Blocks

## Primo programma

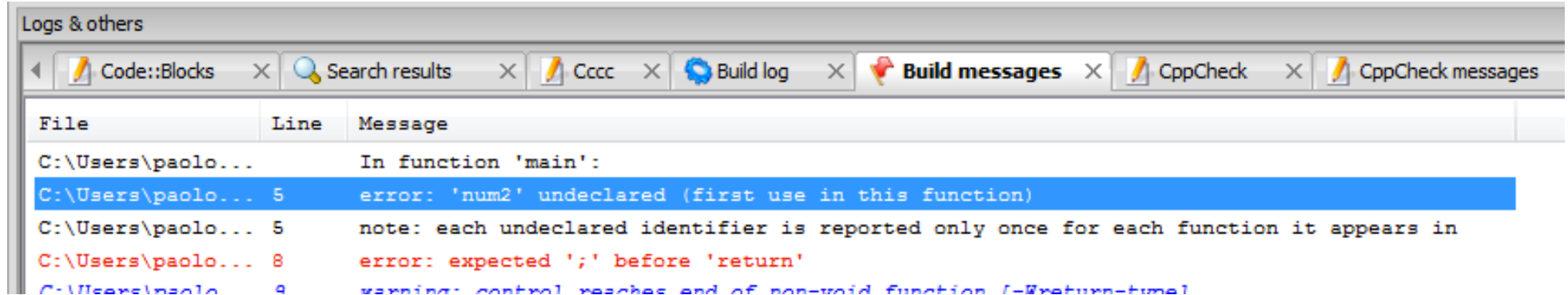
Scrivere un programma che prende in ingresso due numeri interi, calcola la loro somma e stampa a video il risultato.

```
#include <stdio.h>
int main(){
    int num1, risultato;
    scanf("%d", &num1);
    scanf("%d", &num2);
    risultato = num1 + num2;
    printf("%d", risultato)
    return 0;
}
```

*Proviamo a compilare! Cosa ci dice il compilatore?*

# Code:Blocks

## Primo programma



- Il primo errore “num2” undeclared indica che la **variabile num2 non** è mai stata **dichiarata**
- Il secondo errore indica invece che manca un ;

Cliccando sulla riga relativa all'errore, nel codice viene indicata la riga contenente lo stesso.

# Code:Blocks

## Primo programma

Dov'è l'errore questa volta?

```
#include <stdio.h>
int main()
    int lnum, num2,risultato;
    scanf("%d", &lnum);
    scanf("%d", &NUM2);
    risultato = num1 + num2;
    printf("%d", risultato);
    return 0;
}
```



# Code:Blocks

## Primo programma

```
error: invalid suffix "num" on integer constant
```

Una variabile non può iniziare con una cifra.

Proviamo un'altra volta: dov'è l'errore ora?

```
#include <stdio.h>
int main(){
    int num1, num2, risultato;
    scanf("%d", num1);
    scanf("%d", &num2);
    risultato = num1 + num2 ;
    printf("%d", risultato);
    return 0;
}
```

# Code:Blocks

## Primo programma

Il compilatore non da un vero e proprio errore, ma eseguendo il programma:

Windows: è possibile ricercare online una soluzione al problema.

→ Cerca una soluzione online e chiudi il programma

→ Chiudi il programma

→ Esegui debug del programma

▼ Visualizza dettagli problema

Il problema è dato dalla mancanza dell'operatore unario di indirizzo &

- La maggior parte del tempo la si passa a correggere errori:  
è perfettamente normale!
- Sperimentare fa parte della programmazione
- Leggere e interpretare i messaggi del compilatore
- Commentare il codice
- Tenere manuale sotto mano
- In caso di dubbi... chiedere!