



**UNIMORE**  
UNIVERSITÀ DEGLI STUDI DI  
MODENA E REGGIO EMILIA

# Dispense del Corso di Laboratorio di Fondamenti di Informatica II e Lab

## **Un Breve Ripasso**

Ultimo aggiornamento: 20/02/2019

# La Funzione `main()`

- La funzione *main* di un programma C è uguale a tutte le altre funzioni, tranne che:
  - ha un nome convenzionale, fissato;
  - è la funzione invocata per far partire il programma;
  - deve sempre essere presente in un programma C;
- **Chi la invoca?**
- Essendo una funzione come le altre, **può avere dei parametri?**

# Argomento da Linea di Comando

- La funzione main:
  - è invocata dal sistema operativo
  - come parametro può ricevere un array di stringhe che corrisponde all'elenco degli argomenti scritto dall'utente a linea di comando.
- Esempio di invocazione da linea di comando del programma nomeprog:

nomeprog pippo 12 pluto 25

nome del programma                      argomenti (4)

# Argomento da Linea di Comando

- I parametri della funzione `main` sono 2:
  - ***int argc***: contiene il numero di stringhe inserite dall'utente a linea di comando (cardinalità del 2° argomento). Attenzione: anche il nome del programma è una stringa inserita dall'utente e viene conteggiata;
  - ***char \*argv[]***: l'array che contiene le stringhe inserite dall'utente a linea di comando. Ogni elemento dell'array è un puntatore a carattere.

# Argomento da Linea di Comando

- Quindi l'interfaccia completa della funzione `main` è la seguente:

**`int main (int argc, char *argv[])`**

- Se non servono `argc` e `argv` l'interfaccia assume la forma semplificata:

**`int main ()`**

- Il valore di ritorno della funzione `main` può essere usato per restituire un codice numerico di errore al sistema operativo

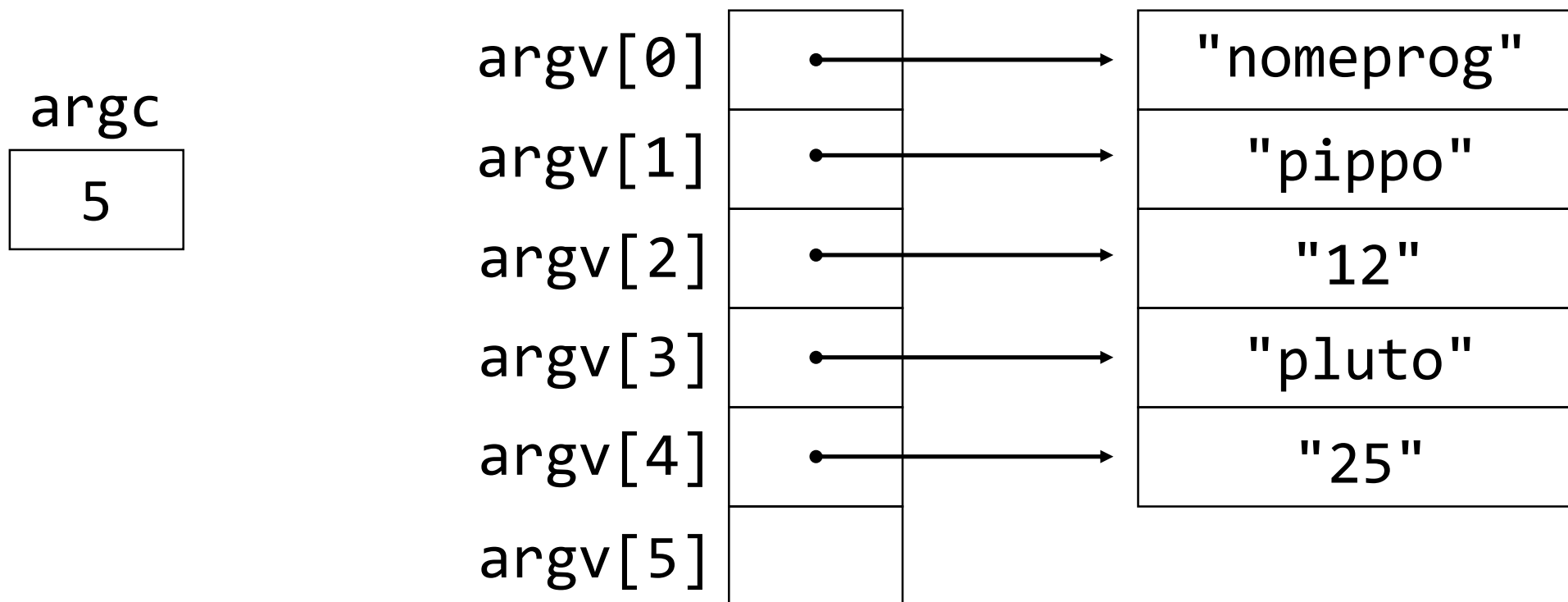
# Argomento da Linea di Comando

- Tornando all'esempio precedente:

`nomeprog pippo 12 pluto 25`

- `argv[0]` -> **nome del programma**, nell'esempio `nomeprog`
- `argv[1]` -> **1° argomento**, nell'esempio `pippo`
- `argv[2]` -> **2° argomento**, nell'esempio `12`
- ...
- `argv[argc-1]` -> **ultimo argomento**, nell'esempio `25`
- `argv[argc]` -> per convenzione è NULL

# Argomento da Linea di Comando



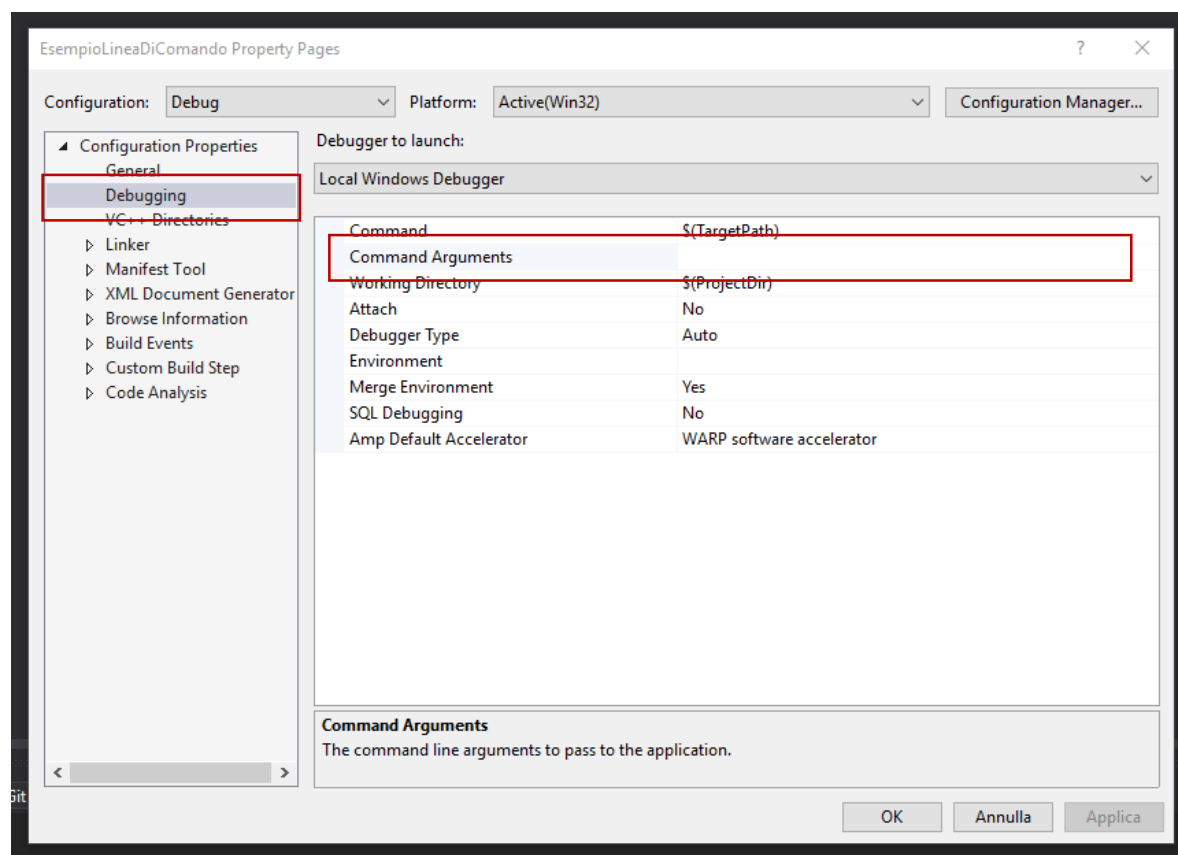
# Argomento da Linea di Comando (6)

- Come passare argomenti dalla linea di comando *quando non c'è una linea di comando*, come negli ambienti di sviluppo integrati (ad esempio Visual Studio)?
- *Esiste un'apposita opzione da menu!*



# Argomento da Linea di Comando

- In Visual Studio 2017: Alt+F7 -> Debugging -> Command Arguments

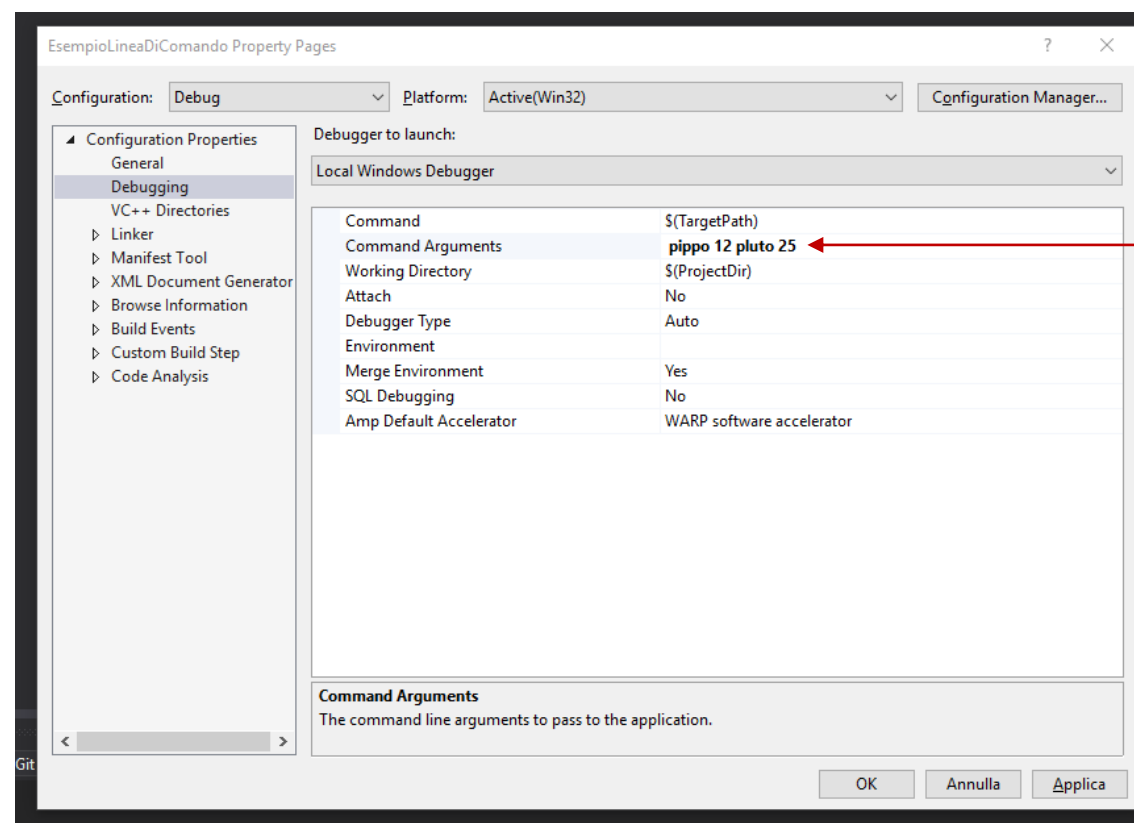


# Argomento da Linea di Comando

- In questo caso il nome del programma viene aggiunto automaticamente da Visual Studio, **NON** bisogna quindi specificarlo nella lista dei *Command Arguments*.
- I parametri da linea di comando devono essere separati da spazi.
- Se voglio inserire un parametro contenente spazi devo usare le virgolette.
- Se ad esempio se voglio che uno dei parametri sia *ciao mamma* dovrò scrivere "ciao mamma" nella lista dei parametri.

# Argomento da Linea di Comando

- Tornando all'esempio precedente: nomeprog pippo 12 pluto 25



# Note sull'Esecuzione di un Programma

- Normalmente la console di Visual Studio si chiude al termine dell'esecuzione del programma, impedendoci così di vedere l'output (cioè che è stato scritto su *standard output*).
- Per ovviare a questo problema potete eseguire il programma premendo CTRL + F5. NON aggiunte *system("pause")* o *getchar()*.
- Se anche avviando il programma con CTRL + F5 la console si chiude al termine dell'esecuzione è perché il progetto di Visual Studio è stato creato con impostazioni "scorrette".
- Per modificare le impostazioni potete aprire le proprietà del progetto (ALT + F7 o tasto destro sul nome del progetto -> *properties*) quindi cercare *Linker* -> *System* -> *SubSystem* e scegliere la voce *Console (/SUBSYSTEM:CONSOLE)* dal menu a tendina.

# Note sull'Esecuzione di un Programma

