

# 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

## 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 scritti dall'utente a linea di comando.
- Esempio di invocazione da linea di comando del programma nomeprog

*C:> nomeprog pippo 12 pluto 25*

Nome programma      Argomenti n. 1, n. 2, n. 3, n. 4

# Argomenti 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)
- **char \*argv[]**: l'array che contiene le stringhe inserite dall'utente a linea di comando (ogni elemento dell'array è un puntatore a carattere)

# Argomenti 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, già nota:

**int main ()**

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

# Argomenti da linea di comando

- `argv[0]` è la prima stringa inserita a linea di comando, quindi il nome del programma

`argv[0]` → **nome del programma**, nell'es. “nomepreog”

`argv[1]` → **1° argomento**, nell'es. “pippo”

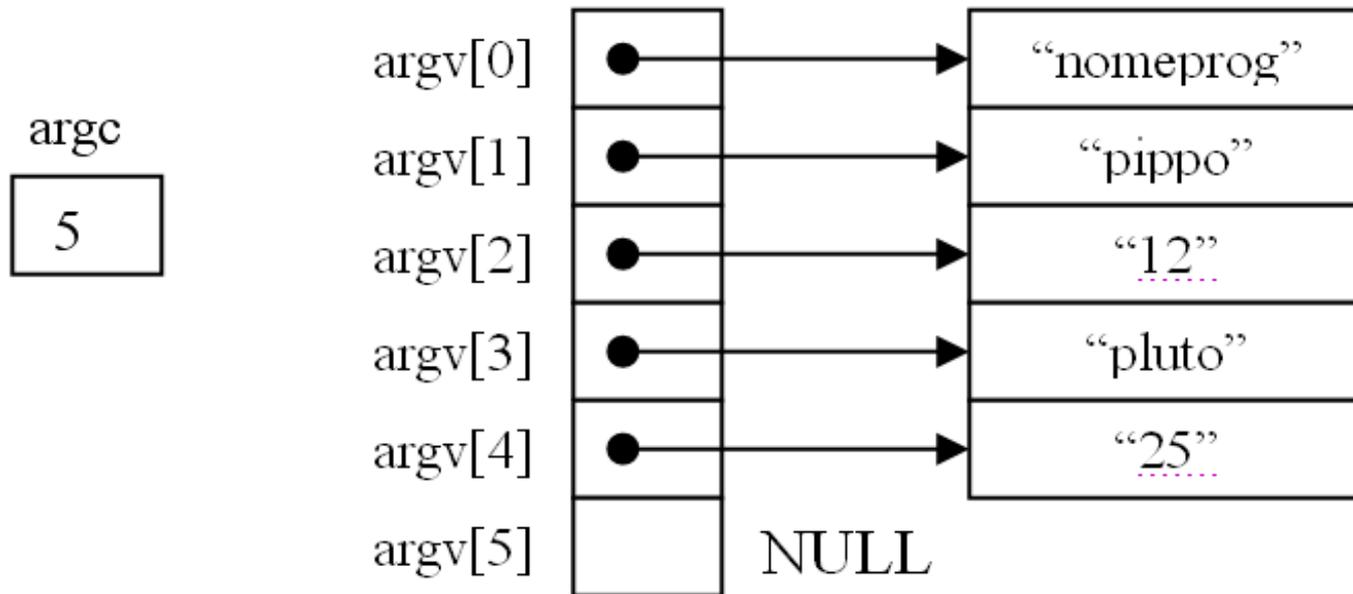
`argv[2]` → **2° argomento**, nell'es. “12”

...

`argv[argc-1]` → **ultimo argomento**, nell'es. “25”

`argv[argc]` → per convenzione è NULL

# Argomenti da linea di comando



# Argomenti da linea di comando

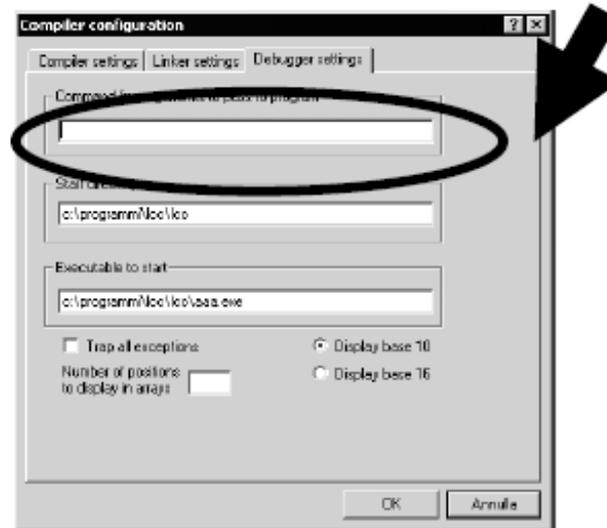
- Come passare argomenti dalla linea di comando ... quando non c'è una linea di comando, come negli ambienti di sviluppo integrati (esempio lcc)?

*C:> nomeprog pippo 12 pluto 25*

- *Esiste un'apposita opzione da menu*

# Argomenti da linea di comando

**In lcc:** Compiler / Settings pagina Debugger Settings



# Esempio 1

- Scrivere un programma che dato un carattere e una stringa da linea di comando, conti quante volte il carattere compare nella stringa

*c:> contaOccorrenze c stringaincuicercare*

- Algoritmo
  - Recuperare gli argomenti
  - Scandire la stringa carattere per carattere, contando le occorrenze del carattere dato
  - Scrivere il numero di occorrenze a video

# Esempio 1

```
#include <stdio.h>
```

```
int main (int argc, char* argv[]) {  
    int cont=0, i=0;  
    char ch=argv[1][0];  
    char *s=argv[2];  
    while (s[i] !='\0')  
        if (s[i++] ==ch)    cont++;  
    printf (“totale occorrenze: %d”, cont);  
}
```

# Esempio 2

- Scrivere un programma che sommi tutti i numeri passati come argomenti a linea di comando e stampi a video questo risultato.
- Algoritmo
  - Recuperare gli argomenti
  - Trasformare ciascun argomento (stringa) in un intero: si utilizza una funzione di libreria *int atoi(char \*)*
  - Scrivere il risultato a video

# Esempio 2

```
#include <stdio.h>
```

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

```
    int sum=0, i=0;
```

```
    while (i < (argc-1)) sum += atoi(argv  
    [++i]);
```

```
    printf (“somma totale : %d”, sum);
```

```
}
```