

Il linguaggio C

Un corso full-immersion

Il primo (storico) programma C

```
#include <stdio.h>
main() {
    printf("Hello World!");
}
```

Il primo (storico) programma C

```
#include <stdio.h>

main() {
    printf("Hello World!");
    system("pause");
}
```

Ma non è successo nulla!

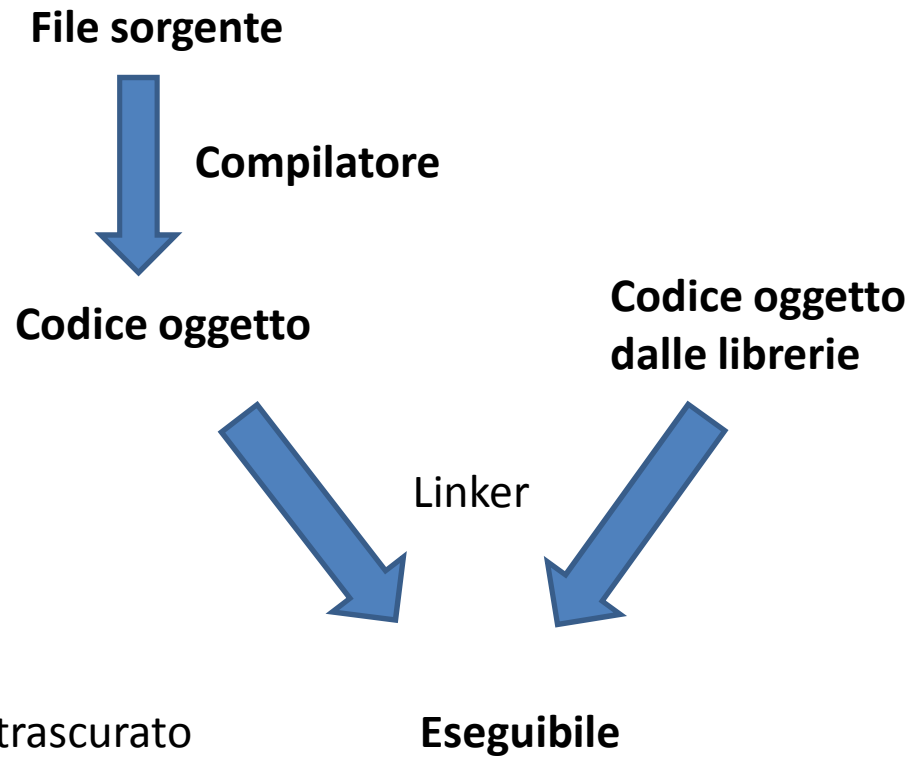
Sbagliato, in realtà è successo qualcosa...

Però è scomparso *molto* rapidamente. Aggiungiamo una istruzione per far restare il calcolatore “in attesa”

Cosa è successo?

- Premendo il pulsante “Compila ed esegui” sono successe un sacco di cose.
 - Il file sorgente è stato salvato (è un semplice file di testo!)
 - Il compilatore lo prende e lo trasforma in codice oggetto
 - Il linker prende il codice oggetto del nostro programma, delle librerie e collega tutto insieme ottenendo un file eseguibile
 - Infine, viene chiesto al sistema operativo di eseguire quel programma.

Cosa è successo? (Cont)



In realtà abbiamo trascurato qualche particolare...

Cosa è successo? (cont.)

- Dev-c è soltanto un IDE (*Integrated Development Environment*), vale a dire un blocco-note con funzioni aggiuntive utili per scrivere programmi
- Il compilatore e il linker sono quelli del progetto *gcc*

Multilingue

- In svedese *Hello world* si dice: “*Hej världen!*”.
- Esercizio: modificare il programma precedente perché scriva i saluti in entrambe le lingue, una sotto l'altra.

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

```
main() {  
    printf("Hello World!");  
    printf("Hello Varlden!");  
    system("pause");  
}
```

No, così non funziona!


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

```
main() {
```

```
    printf("Hello World!\n");
```

```
    printf("Hello Varlden!\n");
```

```
    system("pause");
```

```
}
```

Adesso sì!

- La sequenza `\n` è una sequenza speciale.
Rappresenta il carattere di a-capo (infatti la `n` sta per newline!)
- Esistono altre sequenze di questo tipo
 - compito a casa: cercarle! (Suggerimento: cercare *escape sequence*)

- La sequenza `\n` è una sequenza speciale.
Rappresenta il carattere di a-capo (infatti la `n` sta per newline!)
- Esistono altre sequenze di questo tipo
 - compito a casa: cercarle! (Suggerimento: cercare *escape sequence*)

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



Perché qui no?

```
main() {
```



```
    printf("Hello World!\n");
```

```
    printf("Hello Varlden!\n");
```

```
    system("pause");
```



```
}
```

In C ogni istruzione deve essere terminata dal punto-e-virgola, altrimenti il compilatore si arrabbia, e di brutto.

`#include <stdio.h>`

- Questa a stretto rigore non è una istruzione C.
- Tecnicamente si chiama “direttiva per il precompilatore”
- Serve a un programma che si chiama, appunto, precompilatore: esso sostituisce questa linea facendo copia e incolla del file `stdio.h`
- Esistono altre direttive per il precompilatore: nessuna è terminata dal punto-e-virgola!

```
main() {  
    printf("Hello World!\n") ;  
    printf("Hello Varlden!\n") ;  
    system("pause");  
}
```

Esistono delle precise motivazioni per cui non ci voglia qui il punto-e-virgola.

Per ora assumete soltanto: dopo una parentesi graffa non ci va.

Variabili

- Il nome
 - Univoco
 - Case sensitive
 - Per convenzione è in minuscolo
 - Composto di caratteri alfanumerici (lettere, numeri, underscore _)
 - Inizia con un carattere alfabetico
 - Non può essere una keyword
- Il tipo
 - Semplice
 - Strutturato
 - Array
 - Struct
- Il valore

Le keyword del C

Queste parole hanno un significato speciale e ben preciso nel linguaggio.

Non è possibile sceglierle come nomi di variabile! (E neppure di funzione, come vedremo...)

```
auto break case char const continue default do  
double else enum extern float for goto if  
int long register return short signed sizeof static  
struct switch typedef union unsigned void volatile  
while
```


Esempi di dichiarazioni

```
int i;
```

```
float speed;
```

```
int altezza = 130;
```

```
char c = 'D';
```

```
double larghezza, lunghezza, profondita;
```

```
double larghezza;
```

```
double lunghezza;
```

```
double profondita;
```

Tipo	byte	bit	Range
int	4	32	-2,147,483,648 -> +2,147,483,647
char	1	8	-128 -> +127
float	4	32	
double	8	64	

Ma il char contiene un
carattere oppure un
numero?

Usiamo le variabili

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

```
main() {
```

```
    int a,b, c, media;
```

```
    a = 5;
```

```
    b = 7;
```

```
    c = 11;
```

```
    media = (a+b+c)/3;
```

```
    printf("La media tra %d e %d e %d fa %d", a, b, c, media);
```

```
    system("pause");
```

```
}
```

Cosa fa questo programma?

Notare la sintassi particolare dell'istruzione printf!

Usiamo le variabili

```
#include <stdio.h>
main() {
    float a,b, c, media;
    a = 5;
    b = 7;
    c = 11;
    media = (a+b+c)/3;
    printf("La media tra %f e %f e %f fa %f", a, b, c, media);
    system("pause");
}
```

Cosa fa questo programma?

Notare la sintassi particolare dell'istruzione printf!