

---

# #Direttive per il Preprocessore



# #include

Rende disponibili al programma tutte le funzioni e le costanti dichiarate nel file header passato.

Es:

```
#include <math.h> //permette di utilizzare le funzioni matematiche: sqrt(), pow(),...
```

o

```
#include "myLib.h" //cerca il file header personalizzato nella stessa cartella del sorgente .c
```



# Header File

Un header file (file di testo con estensione .h) contiene un qualsiasi numero di definizioni di costanti (dette **MACRO**) e di **prototipi di funzione** (la sintassi d'uso della funzione).

I prototipi permettono al compilatore di produrre un **codice oggetto** che può essere facilmente unito ("linkato") con quello della relativa libreria in futuro, anche senza avere la libreria sottomano al momento.

Una volta lanciata la compilazione del sorgente con il main(), il preprocessore **sostituisce** alla riga con l'**include** i prototipi delle funzioni e tutte le altre direttive che trova nel file header (si dice che Espande le Direttive) e la compilazione continua.

## #define

Permette definire delle MACRO, ovvero costanti o funzioni che consentono di parametrizzare il sorgente (Tecnica delle Costanti Manifeste) o rendere più leggibile il codice.

Es:

```
#define MAX_NUM      100 //Da qui in poi al posto di scrivere 100 useremo MAX_NUM  
#define AVVISO       printf("\nFino a qui, funziona in modo corretto");  
#define STAMPA(a)     printf("\nLa variabile vale %d",a);  
#define MOLTIPLICA(a,b) (a) * (b)
```

Convenzionalmente i nomi delle macro vengono scritti con delle lettere MAIUSCOLE.

Come per include, il preprocessore legge la definizione della MACRO e, ogni volta che ne incontra il nome all'interno del file sorgente SOSTITUISCE al simbolo il corrispondente valore, SENZA verificare la correttezza sintattica dell'espressione risultante.



# MACRO predefinite

Tipicamente sono predefinite 5 macro standard:

- \_LINE\_** Valore decimale del numero della linea corrente del sorgente.
- \_FILE\_** Stringa del nome del file in corso di compilazione.
- \_DATE\_** Stringa della data di compilazione (formato Mmm dd yyyy).
- \_TIME\_** Stringa dell'ora di compilazione (formato hh:mm:ss).
- \_STDC\_** Contiene il valore 1 se il compilatore e' conforme allo standard ANSI.

Ma per conoscere tutte quelle predefinite sul compilatore installato:

```
C:> echo | gcc -dM -E -
```

(solo in Windows)

Istruisce il preprocessore su quali porzioni di codice passare o meno al COMPILATORE. A seconda del verificarsi di CONDIZIONI parti di codice possono essere scartate (come per i commenti).

## **#ifdef      #if      #defined**

Es:

```
#if defined (MACRO)
    int num = MACRO;
#else
    int n;
#endif
```

Solo una delle due dichiarazioni sarà compilata, a seconda che MACRO sia stata precedentemente definita o meno.