

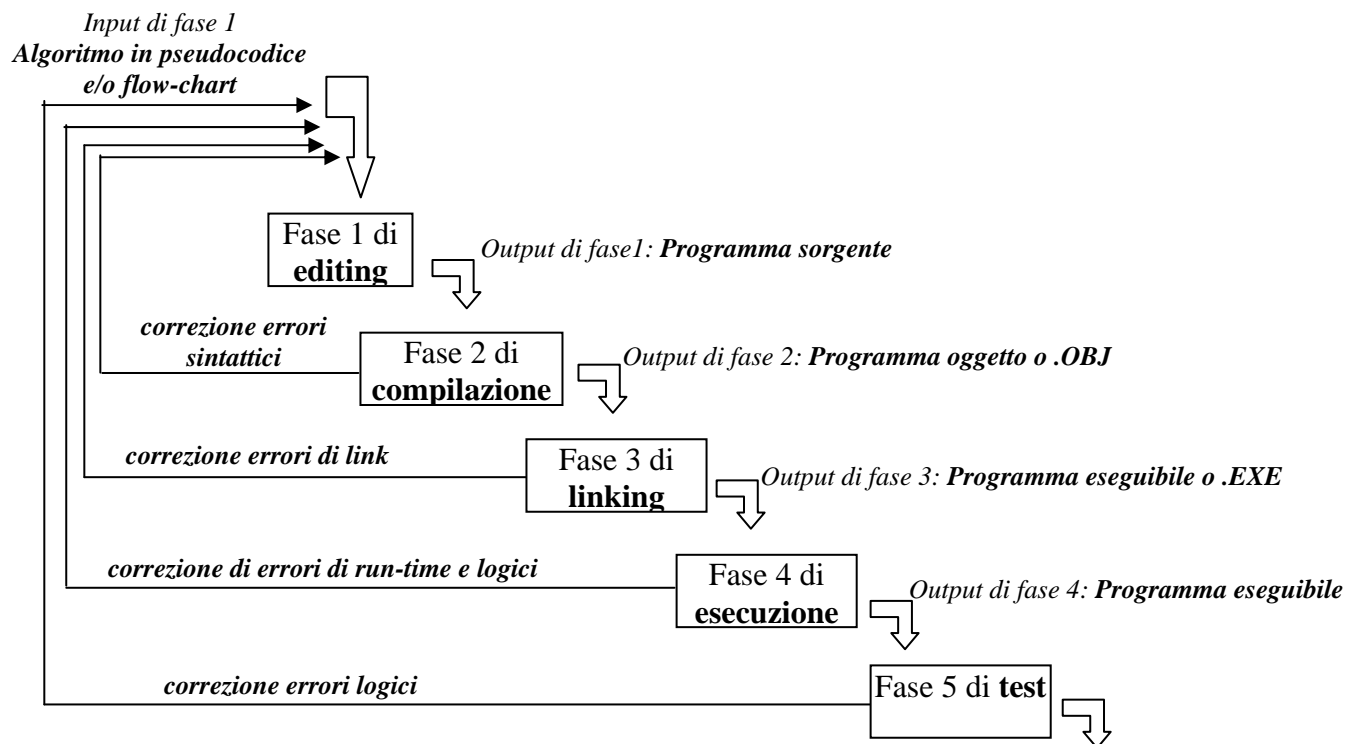
## 5. COMPILATORI ED INTERPRETI

Sono entrambi programmi in grado di *tradurre* le istruzioni di un programma scritto in un linguaggio di programmazione ad alto livello detto **programma sorgente**, in istruzioni comprensibili al computer ossia in istruzioni di un programma scritto in **linguaggio macchina** detto **programma oggetto** (sequenze di '0' ed '1' comprensibili per l'elaboratore).

Sono in pratica dei programmi **traduttori** ma sono profondamente diversi nel funzionamento perché:

- I **compilatori** sono programmi che accettano in input un intero programma scritto in linguaggio ad alto livello (*programma sorgente*) e lo traducono interamente in un programma scritto in linguaggio macchina (*programma oggetto*);
- Gli **interpreti** sono programmi che accettano in input le singole istruzioni di un programma scritto in linguaggio ad alto livello e le traducono una alla volta in singole istruzioni in linguaggio macchina.

### Schema delle fasi di compilazione di un programma sorgente



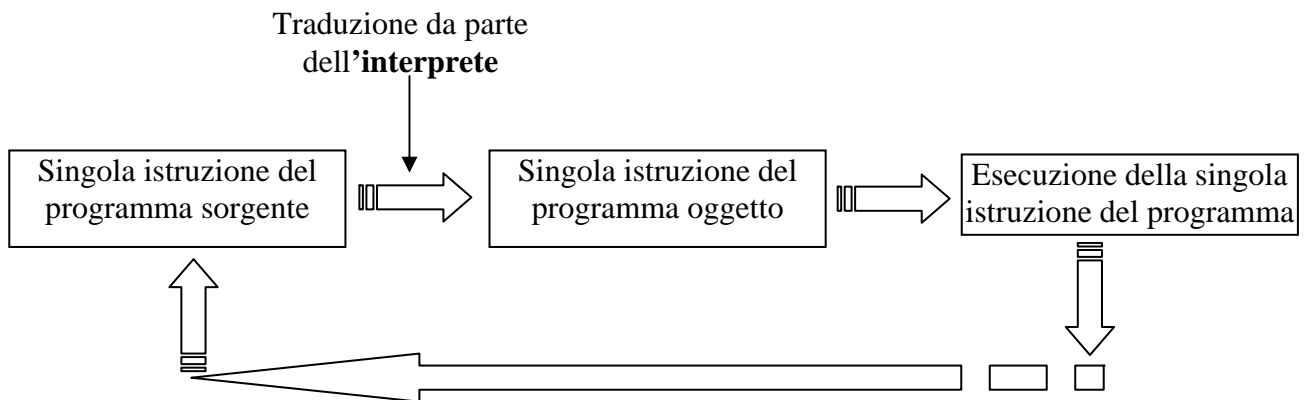
**Programma compilatore o "compiler"** è il programma che traduce il programma sorgente compiendo l'analisi lessicale, sintattica e semantica

**Programma di link o 'linker'** è il programma che collega il programma oggetto con altri moduli oggetto già compilati del sistema operativo o built-in del linguaggio stesso (per chiamate a funzioni e/o utilizzo di variabili e costanti di sistema o del linguaggio di programmazione stesso)

Output di fase 5: **Programma definitivo corretto**

**Programma eseguibile o .EXE** è il programma oggetto completo dei moduli di sistema o built-in invocati

Schema della fase di interpretazione di un programma sorgente



*L'interprete traduce la prima istruzione del programma e la esegue;  
 L'interprete traduce la seconda istruzione del programma e la esegue;  
 .....  
 L'interprete traduce l'ultima istruzione del programma e la esegue.*

Compilatori ed interpreti:vantaggi e svantaggi

|                  | <b>COMPILATORE</b>   | <b>INTERPRETE</b>   |
|------------------|--|---|
| <b>vantaggi</b>  | Il programma sorgente viene tradotto una volta sola                                    | Il programma sorgente può essere modificato interattivamente riducendo il debugging |
|                  | L'esecuzione del programma è più veloce in quanto la fase di traduzione è già avvenuta |   |
|                  | Il programma compilatore non deve risiedere in memoria centrale                        |   |
|                  | E' possibile garantire la segretezza del programma sorgente (delle linee di codice)    |   |
| <b>svantaggi</b> | Ogni volta che si modifica il programma sorgente occorre ricompilare                   | Il programma sorgente viene tradotto ogni volta che deve essere eseguito            |
|                  |  | L'esecuzione del programma è più lenta in quanto avviene assieme alla traduzione    |
|                  |  | Il programma interprete deve risiedere in memoria centrale                          |
|                  |  | Non è possibile garantire la segretezza del programma sorgente                      |

## IL SOFTWARE

Con il termine **software** si intende quell'insieme di programmi che consentono di gestire ed utilizzare il computer stesso. Costituisce *l'aggancio* tra computer ed utente.

Distinguiamo due grandi categorie di software:

1. **IL SOFTWARE APPLICATIVO:** ossia l'insieme di tutti quei programmi che si occupano di automatizzare il lavoro degli utenti in determinati campi di applicazione lavorativi e/o di svago (Es. WORD PROCESSOR, FOGLI ELETTRONICI, VIDEOGIOCHI, PROGRAMMI PER LA GESTIONE DELLE IMMAGINI, etc.);
2. **IL SOFTWARE DI BASE:** ossia l'insieme di tutti quei programmi connessi direttamente al funzionamento della macchina ed alla gestione delle sue periferiche (Es. SISTEMA OPERATIVO, gli ambienti di sviluppo integrati\*)

N.B. Il **sistema operativo** è un insieme di programmi che gestiscono il computer e rendono possibile la relazione uomo-macchina. Esso consente anche una visione astratta delle risorse del sistema ed una loro corretta gestione, sicura ed efficace in modo completamente svincolato dalla conoscenza delle caratteristiche hardware.

Il software rispetto alle modalità di pagamento può suddividersi in:

- **STANDARD**                   ossia a pagamento;
- **FREEWARE**               completamente gratuito e completamente funzionante (spesso viene richiesto all'utilizzatore il versamento di un contributo non obbligatorio volontario nel caso in cui il programma piaccia);
- **SHAREWARE**           il software viene offerto in "parte" all'utilizzatore (*share* = parte, quota), nel senso che non offre la potenzialità di tutte le sue caratteristiche e ne viene consentito l'utilizzo per un periodo di tempo limitato (di solito non superiore ai 30 giorni);
- **OPEN SOURCE:**       il software è completamente gratuito e consegnato con i sorgenti pubblici modificabili con l'implicazione di mettere a disposizione di tutti gli altri fruitori (la *community*) gli eventuali aggiustamenti o sviluppi.

Il software è caratterizzato in genere da una coppia di numeri del tipo

N.M

dove:

- N è la **VERSIONE** o **RELEASE**                   ossia l'implementazione di nuove caratteristiche non disponibili nella versione precedente;
- M è la **REVISIONE**                               ossia la correzione di errori o bug della stessa versione senza implementare nuove funzionalità.

(\*) Gli **ambienti di sviluppo integrati** o **IDE** – Integrated development Environment – sono un insieme di programmi relativi ad un determinato linguaggio di programmazione che permettono di scrivere, correggere, compilare, debuggare, e testare i programmi nel linguaggio di programmazione scelto senza ricorrere alle funzionalità del sistema operativo.