

## LINGUAGGI TRADIZIONALI

**Linguaggi di programmazione.** Qualunque attività del computer consiste in una serie ordinata di operazioni detta *programma*. Esso deve essere scritto in un linguaggio (chiamato *linguaggio di programmazione*) destinato alla comunicazione tra uomo e computer. La comprensione del programma da parte del computer deve essere perfetta, perché solo allora è eseguito correttamente. La sintassi del linguaggio deve obbedire a regole universalmente condivise, codificate in apposite convenzioni internazionali. La struttura dei linguaggi è protetta da appositi standard, redatti, aggiornati e conservati da organismi quali l'ANSI (American National Standards Institute).

**La traduzione del programma.** Il computer è in grado di comprendere e eseguire solo poche operazioni elementari e solo se sono scritte nel suo linguaggio, chiamato *linguaggio macchina*, che consiste in lunghe serie di zeri e uno (istruzioni binarie). L'uomo, da parte sua, preferisce esprimersi in un linguaggio più vicino a quello parlato, cioè in un *linguaggio di programmazione evoluto*. Il processo di traduzione tra i due linguaggi (*compilazione*) è affidato al computer stesso, dotato perciò di opportuni *programmi compilatori*. In seguito c'è la fase del *lincaggio* (dall'inglese *to link*) effettuata da un *programma linker*, in cui al programma tradotto sono aggiunti sottoprogrammi preesistenti (es. per l'input/output) conservati in apposite librerie (fig. A). Alcuni linguaggi non sono associati a un processo di compilazione ma a un processo di *interpretazione*, in cui ogni istruzione è tradotta ed eseguita al momento.

**Portabilità dei programmi.** Ogni computer ha un suo specifico linguaggio macchina che non funziona su un computer di un'altra marca. Il vantaggio dell'uso di un linguaggio di programmazione evoluto, indipendente dalla macchina, consiste nella possibilità di *trasferire* programmi da una macchina all'altra. Questa proprietà dei programmi si chiama *portabilità*.

**Linguaggi tradizionali.** ● *Algol*: è usato per descrivere algoritmi di analisi numerica. Ha soprattutto rilevanza storica: è stato il primo linguaggio definito in modo formale nel 1960. ● *Assembler*: è in corrispondenza uno-a-uno col linguaggio macchina: a ogni istruzione del linguaggio macchina corrisponde un'istruzione assembler. È più leggibile del linguaggio macchina perché permette di sostituire gli zeri e gli uno con *codici mnemonici* con i quali designare istruzioni e dati (fig. B). ● *Basic*: è stato molto usato in ambito scientifico e tecnico. Spesso il Basic viene interpretato e non compilato. ● *C* e *C++*: il C, e la sua versione più potente C++, orientata agli oggetti, sono usati soprattutto da chi sviluppa software di sistema e software tecnico (fig. C). ● *Cobol*: è tra i linguaggi più diffusi al mondo nelle applicazioni gestionali (contabilità, paghe, magazzini ecc.). Molti package applicativi presenti sul mercato sono scritti in Cobol. ● *Fortran*: è il più antico linguaggio evoluto (1958) ed è usato nella programmazione scientifica e tecnica (programmazione lineare, routine matematiche ecc.) (fig. D). ● *Lisp*: è stato disegnato per elaborare dati non numerici, successioni arbitrarie di caratteri alfabetici; è usato principalmente nelle applicazioni di intelligenza artificiale. ● *Pascal*: è un linguaggio nato e usato per scopi didattici; è stato tra i primi *linguaggi strutturati* e ha introdotto molti vincoli nella programmazione che garantiscono la qualità di un programma (fig. E). ● *PLI*: è stato concepito per sostituire Cobol e Fortran, integrando in un solo linguaggio le potenzialità applicative dei due linguaggi. Pur non avendo conseguito l'obiettivo, ha avuto una diffusione notevole e il suo patrimonio applicativo è rilevante.

