

## 7.1 Il concetto di sottoprogramma

Un programma è formato da elementi connessi in modo da raggiungere un determinato scopo. Le istruzioni possono essere considerate i componenti di un programma. Ciascuna istruzione corrisponde a un'azione elementare: ponendo le istruzioni in un determinato ordine il programma svolge il compito cui era stato destinato dal progettista. Da questo punto di vista, un programma non è diverso da un qualunque altro sistema; per esempio un personal computer è costituito da più elementi: la tastiera, il video, l'unità centrale, la stampante ecc., ognuno dei quali è connesso all'altro in un ordine specifico ed è preposto a uno specifico compito. A sua volta, ogni elemento potrebbe essere scomposto in ulteriori componenti. Se cominciassimo a smontare una stampante tra i pezzi componenti troveremmo una consolle di comando, il trattore della carta, la testina di stampa, il motore e così via. Se continuassimo a smontare la stampante (probabilmente distruggendola!) individueremmo centinaia di altri componenti prima di giungere ai pezzi non ulteriormente smontabili.

Un programma non può essere “smontato” oltre il limite delle singole istruzioni, ma è possibile aggregare gruppi di istruzioni per formare dei “semilavorati” detti *sottoprogrammi*. Come un personal computer è composto da tastiera, video, stampante e unità centrale, così un programma per il calcolo degli stipendi potrebbe essere scomposto nei sottoprogrammi di immissione delle ore lavorate, di calcolo dello stipendio e di visualizzazione e stampa della situazione contabile di ogni impiegato.

I sottoprogrammi si usano anche per evitare di replicare porzioni di codice sorgente: invocare un sottoprogramma significa mandare in esecuzione la porzione di codice corrispondente. Se un sottoprogramma è invocato più volte, la porzione di codice è eseguita più volte, tante quante sono le invocazioni. Il vantaggio dei sottoprogrammi è appunto di consentire al programmatore di avere tante chiamate ma una sola porzione di codice.

È possibile poi creare delle librerie, cioè delle raccolte di sottoprogrammi che possono essere utilizzati senza essere a conoscenza dei dettagli implementativi. È quanto avviene con le funzioni `printf()` e `scanf()`, la cui dichiarazione è contenuta nel file `stdio.h`.