

2.1 L'istruzione

Quando si desidera eseguire un'istruzione al verificarsi di una certa condizione, si utilizza l'istruzione `if`. Per esempio, se si vuole visualizzare il messaggio minore di 100 solamente nel caso in cui il valore della variabile intera `i` è minore di 100, si scrive

```
if(i<100) printf("minore di 100");
```

La sintassi dell'istruzione `if` è

```
if(espressione) istruzione
```

dove la valutazione di *espressione* controlla l'esecuzione di *istruzione*: se *espressione* è vera viene eseguita *istruzione*. L'espressione `i<100` è la condizione logica che controlla l'istruzione di stampa e pertanto la sua valutazione potrà restituire soltanto uno dei due valori booleani vero o falso, che in C corrispondono rispettivamente ai valori interi uno e zero. In C, a differenza che in altri linguaggi come il Pascal, non esistono variabili logiche, per cui *falso* è lo zero e *vero* è ciascun valore diverso da zero (in effetti uno è diverso da zero!). Quindi `i<100` è anche un'espressione intera la cui valutazione restituisce 1 quando la variabile `i` è minore di 100 e 0 quando `i` è maggiore o uguale a 100 ■. L'esempio precedente è funzionalmente identico alla successiva sequenza, dove `a` è ancora una variabile intera:

```
a = i<100;
```

```
if(a!=0)
    printf("minore di 100");
```

L'assegnamento `a=i<100` è del tutto lecito, perché viene valutata l'espressione logica `i<100`, che può restituire 1 (vero) o 0 (falso). Il risultato è dunque un numero intero, che viene assegnato alla variabile `a`.

Se osserviamo la sintassi del costrutto `if`, notiamo che nel primo esempio *espressione* è `i<100` mentre nel secondo è `a!=0`, e in entrambi *istruzione* è `printf()`. L'operatore `!=` corrisponde a *diverso da*, per cui valutare l'espressione `a!=0` significa chiedersi se il valore di `a` è diverso da zero. Ma questo equivale a chiedersi se il valore di `a` è vero, che è un controllo eseguito per default; avremmo quindi potuto scrivere anche semplicemente:

```
a = i<100;
if(a)
    printf("minore di 100");
```

Nei paragrafi successivi esamineremo tutti gli operatori di confronto, riprenderemo il discorso sulle espressioni aritmetiche e logiche e vedremo come si integrano le une nelle altre.

La sintassi completa dell'istruzione `if` è:

```
if(espressione) istruzione1 [else istruzione2]
```

dove la valutazione di *espressione* controlla l'esecuzione di *istruzione1* e quella di *istruzione2*: se *espressione* è vera viene eseguita *istruzione1*, se è falsa viene eseguita *istruzione2* ■.

Nell'esempio precedente è stato omesso il ramo `else`; il fatto è del tutto legittimo poiché tale ramo è opzionale, come evidenziato dalle parentesi quadre della forma sintattica completa e come vedremo anche nel prossimo paragrafo. Modifichiamo ora l'esempio in modo da visualizzare un messaggio anche quando la variabile `i` non è minore di 100 (Listato 2.1).

```
/* Utilizzo if-else */

#include <stdio.h>

main()
{
    int i;

    printf("Dammi un intero: ");
    scanf("%d", &i);

    if(i<100)
        printf("minore di 100\n");
    else
        printf("maggiore o uguale a 100\n");
}
```

Listato 2.1 Esempio di diramazione del flusso di esecuzione

Abbiamo introdotto il ramo `else` dell'istruzione `if`, inserendo l'istruzione che visualizza maggiore o uguale a 100, da eseguire nel caso l'espressione di controllo risulti falsa. Si noti che i rami `if-else` si escludono mutuamente, o viene eseguita la prima `printf` o viene eseguita la seconda, mai entrambe.