RICORSIONE

Funzione ricorsiva: funzione che richiama sé stessa. Esistono due tipi di ricorsione:

- *diretta*: funzione invoca sé stessa senza intermediari
- *indiretta*: se coinvolge più di una funzione

Se si utilizzano le funzione ricorsive il debugger risulta molto utile nella ricerca di errori in quanto permette di vedere tutte le chiamate fatte.

La ricorsione corrisponde a una soluzione concisa ed elegante per alcune tipologie di problemi (non tutti). Alcuni problemi hanno una natura intrinsecamente ricorsiva (fattoriale, fibonacci, hanoi…).

Senza un'istruzione di return si creerebbe una catena infinita di chiamate ricorsive.

```
int fattoriale (int n)
{
    if (n == 0)
        return 1;
    return n * fattoriale(n - 1);
}
```

Cosa succederebbe senza queste istruzioni?

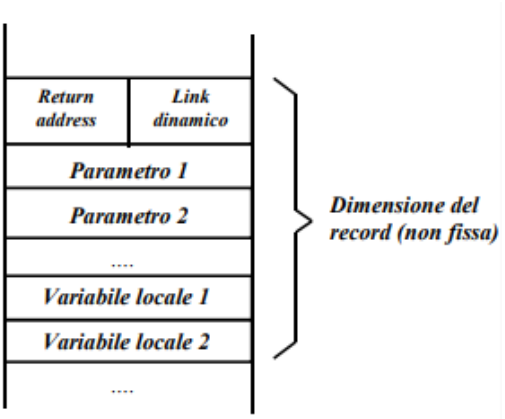
è necessaria quindi la condizione di terminazione che verifica se siamo in presenza di un caso base; identifica la fine della catena ricorsiva.

La ricorsione è controllata da uno o più parametri chiamati argomenti di controllo. In funzione del valore dell'argomento si determina se eseguire un caso base o la chiamata ricorsiva. La sequenza dei valori deve tendere verso uno dei casi base altrimenti la funzione diverge.

Quando una funzione viene invocata, alcune informazioni vengono salvate sullo stack:

- Return address: punto del codice in cui è stata invocata
- Parametri e variabili locali

L'insieme di questi dati sullo stack è detto Record di Attivazione (RdA).



Con la ricorsione ci saranno più istanze della stessa funzione nello stesso istante (instanze multiple).

Iterazione vs ricorsione

- Ricorsione → concisa ed elegante per problemi scomponibili
- Iterazione → meno concisa per lo stesso tipo di problemi, ma più efficiente nello spazio e nel tempo perché non è necessario allocare spazio in memoria per un nuovo record di attivazione ad ogni chiamata ricorsiva