

10.4 La successione di Fibonacci

Nella particolare successione detta di Fibonacci ogni termine è ottenuto sommando i due che lo precedono; il termine generico è pertanto

$$F(n) = F(n-1) + F(n-2)$$

dove n è un numero intero maggiore o uguale a 2. Inoltre $F(0) = 0$ e $F(1) = 1$. Dunque si ha:

$$F(2) = F(2-1) + F(2-2) = F(1) + F(0) = 1+0 = 1$$

$$F(3) = F(3-1) + F(3-2) = F(2) + F(1) = 1+1 = 2$$

$$F(4) = F(4-1) + F(4-2) = F(3) + F(2) = 2+1 = 3$$

$$F(5) = F(5-1) + F(5-2) = F(4) + F(3) = 3+2 = 5$$

$$F(6) = F(6-1) + F(6-2) = F(5) + F(4) = 5+3 = 8$$

$$F(7) = F(7-1) + F(7-2) = F(6) + F(5) = 8+5 = 13$$

...

È evidente il carattere ricorsivo di tale definizione (si veda il Listato 10.4): ogni chiamata di `fibonacci` genera due ulteriori chiamate ricorsive, una passando il parametro attuale $n-1$, l'altra $n-2$; dunque al crescere del valore di n la memoria tende rapidamente a saturarsi.

```
/* Calcolo dei numeri di Fibonacci */

#include <stdio.h>

long int fibonacci(int);

main()
{
    int n;

    printf("Successione di Fibonacci f(0)=1 f(1)=1 f(n)=f(n-1)+f(n-2)");
    printf("\nInserire n: \t");
    scanf("%d", &n);
    printf("Il termine della successione di argomento %d è: %d\n", n, fibonacci(n));
}

long int fibonacci(int n)
{
    if(n==0)        return(0);
    else if(n==1)   return(1);
    else            return(fibonacci(n-1)+fibonacci(n-2));
}
```

Listato 10.4 Calcolo della successione di Fibonacci

✓ NOTA

I programmi che sfruttano la ricorsività sono in larga misura inefficienti, tanto in termini di occupazione di memoria quanto in termini di velocità di esecuzione. La ragione principale è che nelle funzioni ricorsive spesso vengono ripetuti calcoli già eseguiti in precedenza. Si riprenda l'esempio delle combinazioni: nel caso di $\text{comb}(5, 2)$, $\text{comb}(2, 1)$ è calcolato tre volte!

Si può facilmente verificare il peso di queste ripetizioni inserendo, all'interno delle funzioni esaminate, prima della chiamata ricorsiva, una `printf` che visualizzi il valore delle variabili trattate dalle funzioni stesse.