

3.3 Calcolo del fattoriale

Utilizziamo il costrutto `for` per il calcolo del fattoriale, indicato con $n!$, di un intero n , definito da

$$n! = n \cdot (n-1) \cdot (n-2) \cdot (n-3) \dots \cdot 2 \cdot 1$$

dove $1!$ e $0!$ sono per definizione uguali a 1. Avremo, per esempio, che

$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

```
/* Calcolo di n! (n fattoriale) */
#include <stdio.h>

main()
{
    int n, fat, m;

    printf("CALCOLO DI N!\n\n");
    printf("Inser. n: ");
    scanf("%d", &n);

    fat = n;
    for(m=n; m>2; m--)
        fat = fat*(m-1);

    printf("Il fattoriale di: %d ha valore: %d\n", n, fat);
}
```

Listato 3.2 Calcolo del fattoriale di n

Nell'ipotesi di non considerare il caso $n = 0$, un algoritmo possibile è quello del Listato 3.2. Se viene passato in ingresso il valore 4, `fat` assume tale valore:

```
fat = n;
```

Il ciclo `for` inizializza 4 a `m` e controlla che sia maggiore di 2. Viene eseguito una prima volta il ciclo

```
fat = fat*(m-1);
```

e `fat` assume il valore 12. Di nuovo il controllo dell'esecuzione passa al `for` che decrementa il valore di `m` e verifica se $m > 2$, cioè se $3 > 2$. Viene eseguito il corpo del ciclo

```
fat = fat*(m-1);
```

e `fat` assume il valore 24. Il `for` decrementa `m` e verifica se $m > 2$, cioè se $2 > 2$. Questa volta l'esito è negativo e le iterazioni hanno termine. Utilizzando l'operatore `*`, al posto di `fat=fat*(m-1)` avremmo potuto scrivere

```
fat *= m-1;
```

Per considerare anche il caso in cui sia compreso il fattoriale di zero, prima di iniziare il ciclo ci si deve chiedere se n ha tale valore, nel qual caso il risultato è 1.

```
fat = n;
if (n==0)
    fat = 1;
else
    for(m=n; m>2; m--)
        fat = fat*(m-1);
```

L'uso della variabile `m` è necessario perché si desidera mantenere il valore iniziale di `n` per stamparlo nella `printf` finale, altrimenti se ne potrebbe fare a meno utilizzando al suo posto direttamente `n`:

```
fat = n;
if (n==0)
    fat=1;
else
```

```
for(n=n; n>2; n--)  
    fat = fat*(n-1);
```

L'inizializzazione all'interno del `for` `n=n` è del tutto superflua, per cui si può scrivere

```
for(; n>2; n--)  
    fat = fat*(n-1);
```

Questa sintassi è permessa e indica che *esp1* è vuota; il punto e virgola è obbligatorio. Un altro metodo è quello di eseguire le moltiplicazioni successive a partire dal basso: $n! = 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$, inizializzando `fat` a 1 e utilizzando una variabile ausiliaria intera (Listato 3.3). Si noti come con questa soluzione sia già incluso il caso di 0!. Anche questa volta invece di `fat=fat*aux` avremmo potuto scrivere `fat*=aux` ■.

```
/* Calcolo n! (n fattoriale) */  
  
#include <stdio.h>  
  
main()  
{  
    int n, fat, aux;  
  
    printf("CALCOLO DI N!\n\n");  
    printf("Inser. n: ");  
    scanf("%d", &n);  
  
    fat = 1;  
    for(aux=2; aux<=n; aux++) fat = fat*aux;  
  
    printf("Il fattoriale di: %d ha valore: %d\n", n, fat);  
}
```

Listato 3.3 Un'altra possibilità per il calcolo del fattoriale