10.2 Permutazione e disposizioni

In questo paragrafo e nel successivo considereremo alcune procedure di calcolo combinatorio allo scopo di scrivere interessanti funzioni ricorsive.

Si definiscono *permutazioni semplici* di n oggetti distinti i gruppi che si possono formare in modo che ciascuno contenga tutti gli n oggetti dati e che differisca dagli altri soltanto per l'ordine in cui vi compaiono gli oggetti stessi. Dati due oggetti `e1`, `e2` si possono avere solamente due permutazioni; se gli oggetti sono tre le permutazioni semplici diventano sei; se gli oggetti sono quattro (`e1 e2 e3 e4`), si hanno le seguenti 24 possibilità:

```
e1 e2 e3 e4    e1 e2 e4 e3     e2 e1 e3 e4     e1 e2 e4 e3
     e3 e1 e2 e4    e3 e1 e4 e2     e1 e3 e2 e4     e1 e3 e4 e2
     e2 e3 e1 e4    e2 e3 e4 e1     e3 e2 e1 e4     e3 e2 e4 e1
     e1 e4 e2 e3    e1 e4 e3 e2     e2 e4 e1 e3     e2 e4 e3 e1
     e3 e4 e1 e2    e3 e4 e2 e1     e4 e1 e2 e3     e4 e1 e3 e2
     e4 e2 e1 e3    e4 e2 e3 e1     e4 e3 e1 e2     e4 e3 e2 e1
```

In generale, il numero di permutazioni P di n oggetti è dato $P_n = n!$, da cui risultano appunto, nel nostro caso, $4! = 24$ possibilità distinte.

Questo è un problema che abbiamo già risolto. Se desideriamo conoscere il numero di permutazioni di 13 oggetti è sufficiente mandare in esecuzione l'ultimo programma del paragrafo precedente, con l'accortezza di utilizzare un tipo dati adeguato, per scoprire che sono: 6 227 020 800.

Dati n oggetti distinti e detto k un numero intero positivo minore o uguale a n, si chiamano invece *disposizioni semplici* di questi n oggetti i gruppi distinti che si possono formare in modo che ogni gruppo contenga soltanto k oggetti e che differisca dagli altri o per qualche oggetto, o per l'ordine in cui gli oggetti stessi sono disposti. Le disposizioni di quattro oggetti (n=4) presi uno a uno (k=1) sono dunque i gruppi che contengono un solo oggetto:

```
e1      e2      e3      e4
```

cioè in totale quattro. Le disposizioni di quattro oggetti presi due a due (k=2) sono invece 12:

```
e1 e2      e2 e1      e3 e1      e4 e1
     e1 e3      e2 e3      e3 e2      e4 e2
     e1 e4      e2 e4      e3 e4      e4 e3
```

Il calcolo delle disposizioni usa la formula generale:

$$D_{n,k} = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-k+2) \cdot (n-k+1)$$

Nel caso di n=4 e k=1 verifichiamo

$$D_{4,1} = 4$$

e nel caso di n=4 e k=2

$$D_{4,2} = 4 \cdot 3 = 12$$

Possiamo dunque scrivere una procedura ricorsiva che calcoli le disposizioni semplici:

```
int dispo(int k, int n, int m)
     {
     if(n==m-k)
       return(1);
     else
       return(n*dispo(k, n-1, m));
```

```
}
```

Al momento della prima invocazione di `dispo`:

```
dispo(k, n, n);
```

dobbiamo passare alla funzione, oltre ai valori di `k` e di `n`, anche un ulteriore valore `n` (che diventa il parametro formale `m`) perché essa possa conoscere il numero totale degli oggetti e quindi effettuare il controllo `n=m-k`, dato che a ogni ulteriore chiamata il parametro formale `n` viene decrementato di una unità rispetto al precedente. Naturalmente si poteva anche optare per l'utilizzo di una variabile globale `m` inizializzata al valore di `n` (si veda il Listato 10.2).

```
/* Calcolo delle disposizioni semplici di n oggetti presi k a k */

#include<stdio.h>

int dispo(int, int, int);

main()
{
int n, k;

printf("Disposizioni semplici di k su n oggetti\n");
printf("Inser. n: \t");
scanf("%d", &n);
printf("Inser. k: \t");
scanf("%d", &k);
printf("Le dispos. sempl. di %d su %d sono: %d\n", k, n, dispo(k, n, n));
}

int dispo(int k, int n, int m)
{
if(n==m-k)
  return(1);
else
  return(n*dispo(k, n-1, m));
}
```

Listato 10.2 Calcolo delle disposizioni semplici

Osserviamo che il calcolo delle disposizioni è simile a quello del fattoriale. In particolare, le disposizioni di *n* elementi presi *n* a *n* sono proprio pari a *n*!:

$D_{n,n} = n!$

Nel caso fosse n=4 e k=4 avremmo:

$D_{4,4} = 24$

Possiamo allora scrivere una nuova funzione `dispo2()` che sfrutti la funzione `fat` precedentemente definita:

```
/* Calcolo delle disposizioni semplici utilizzando la funzione
       per il calcolo delle permutazioni */

     int dispo2(int k, int n)
     {
     return(fat(n)/fat(n-k));
     }
```

Infatti $D_{n,k}$ = `fat(n)/fat(n-k)`, come si può facilmente dedurre dal confronto delle due formule.