

## 10.3 Combinazioni

Si chiamano *combinazioni semplici* di  $n$  oggetti distinti, presi  $k$  a  $k$  ( $k \leq n$ ) i gruppi di  $k$  oggetti che si possono formare con gli  $n$  oggetti dati, in modo che i gruppi stessi differiscano tra loro almeno per un oggetto. Per esempio, i quattro elementi  $e_1, e_2, e_3$  ed  $e_4$ , presi due a due, danno origine alle seguenti sei combinazioni:

$e_1 e_2$        $e_1 e_3$        $e_1 e_4$        $e_2 e_3$        $e_2 e_4$        $e_3 e_4$

La formula generale che consente di calcolare il numero delle combinazioni è

$$C_{n,k} = D_{n,k}/k!$$

Dunque il numero di combinazioni di  $n$  oggetti presi  $k$  a  $k$  è uguale al numero di disposizioni di  $n$  oggetti presi  $k$  a  $k$ , diviso  $k$  fattoriale.

La funzione `comb`, che calcola il numero di combinazioni semplici possibili, può richiamare `dispo` per calcolare le disposizioni  $D_{n,k}$  e `fat` per calcolare il fattoriale, passando  $k$  come numero di elementi:

```
comb(int k, int n)
{
    return (dispo(k, n) / fat(k));
}
```

Nel Listato 10.3 viene presentato il programma relativo al calcolo delle combinazioni semplici.

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

#include <stdio.h>

int comb(int, int);
int dispo(int, int, int);
int fat(int);

main()
{
    int n, k;

    printf("Combinazioni semplici di k su n oggetti\n");
    printf("Inserire n: \t");
    scanf("%d", &n);
    printf("Inserire k: \t");
    scanf("%d", &k);
    printf("Le combin. sempl. di %d su %d sono: %d\n", k, n, comb(k, n));
}

comb(int k, int n)
{
    return (dispo(k, n, n) / fat(k));
}

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

fat(int n)
{

```

```

if (n==0)
    return(1);
else
    return(n*fat(n-1));
}

```

Listato 10.3 Calcolo delle combinazioni semplici; vengono utilizzate le funzioni `dispo` e `fat` viste precedentemente

Una prima alternativa è quella di utilizzare in `comb` soltanto la funzione `dispo`, dato che  $D_{k,k}$  è uguale a  $k!$ :

```

/* Calcolo delle combinazioni semplici
   utilizzando soltanto la funzione per il
   calcolo delle disposizioni */

comb(int k, int n)
{
    return(dispo(k, n, n)/dispo(k, k, k));
}

```

Una seconda possibilità si ottiene sfruttando la funzione `dispo2` che, come abbiamo visto in precedenza, utilizzava a sua volta `fat` per calcolare le disposizioni:

```

/* Calcolo delle combinazioni semplici utilizzando
   dispo2() e fat() */

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

/* Calcolo delle disposizioni semplici utilizzando fat() */

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

```

Così facendo abbiamo decomposto le formule risolutive di combinazioni e disposizioni rimandando il problema al calcolo del fattoriale. Attenzione, comunque: se  $n$  e  $k$  superano un certo valore, che dipende dalla dimensione degli `int` e dei `long int` dello specifico compilatore, si devono utilizzare funzioni e parametri di tipo `float`.