

9.7 Gestione di una sequenza

Nell'ultimo paragrafo del Capitolo 7 abbiamo esaminato un programma per la gestione di una sequenza tramite un menu con le opzioni di immissione, ordinamento, ricerca completa e ricerca binaria. In quella sede l'array che conteneva la sequenza era una variabile generale cui tutte le funzioni accedevano direttamente.

Adesso presentiamo le modifiche necessarie perché il tutto avvenga mediante un array locale alla funzione `gestione_sequenza` e il passaggio del suo indirizzo alle altre funzioni. Innanzitutto le dichiarazioni devono essere fatte in modo da includere il parametro puntatore all'array:

```
int immissione( int, int * );
void ordinamento( int, int * );
int ricerca( int, int , int * );
int ric_bin( int, int , int * );
void visualizzazione( int, int * );
```

Alle dichiarazioni è stato aggiunto `int *`, per indicare che quel parametro sarà un puntatore a un oggetto di tipo `int`. Supponiamo che in `gestione_sequenza` venga definito l'array `sequenza`:

```
int sequenza[MAX_ELE];
```

Al momento della chiamata delle funzioni tale array deve essere passato come parametro attuale:

```
case 1: n = immissione( n, sequenza );
case 2: ordinamento( n, sequenza );
posizione = ricerca( n, ele, sequenza );
posizione = ric_bin( n, ele, sequenza );
case 5: visualizzazione( n, sequenza );
```

Nella definizione delle funzioni, a sua volta, deve essere esplicitato un nuovo parametro formale:

```
int immissione( int n, int *vet )  
void ordinamento( int n, int *vet )  
int ric_bin( int n, int ele, int *vet )  
void visualizzazione( int n, int *vet )
```

Sorprendentemente, all'interno di ogni funzione, non cambia niente: infatti `vet` è una variabile puntatore all'array: i sottoprogrammi possono accedere all'array e modificare il suo contenuto ma, essendo `vet` una variabile locale, non possono modificarlo.