

14.4 Determinazione del maggiore di una lista

Consideriamo il seguente problema: memorizzare una sequenza di numeri interi terminante con zero in una lista lineare, visualizzare la lista e determinare il maggiore degli elementi. Il valore zero non fa parte della lista.

Dividiamo il problema in tre sottoproblemi:

- • memorizzare la sequenza in una lista lineare;
- • visualizzare la lista;
- • determinare il maggiore della sequenza.

Le dichiarazioni necessarie sono le seguenti:

```
struct elemento {  
    int inf;  
    struct elemento *pun;
```

```
};

elemento *crea_lista2();
void visualizza_lista(struct elemento *);
int maggiore_lista(struct elemento *);
```

Rispetto al problema precedente si dovrà modificare la funzione `crea_lista`, poiché il numero degli elementi della sequenza non è indicato a priori dall'utente; la funzione `maggiore_lista` va sviluppata. Nel Listato 14.2 viene riportata la parte nuova di programma.

Nel resto del capitolo, le funzioni già esaminate precedentemente non verranno ridefinite ma solo dichiarate e invocate.

Evidenziamo le differenze tra `crea_lista2` e `crea_lista` vista anteriormente. Questa volta non possiamo inserire direttamente l'informazione passata dall'utente in un elemento di lista mediante l'istruzione `scanf("%d", &p->inf)`, perché non sappiamo se questa fa parte effettivamente della sequenza o se corrisponde alla marca di fine sequenza (valore zero). A questo scopo utilizziamo la variabile `x` di tipo `struct elemento` per accettare i valori in entrata.

Si richiede all'utente di inserire la prima informazione: se questa è uguale a zero, si assegna a `p` il valore `NULL` e la funzione ha termine:

```
printf("\nInserisci un'informazione (0 per fine lista): ");
scanf("%d", &x.inf);
if(x.inf==0) p = NULL;
```

Nel caso in cui il valore della prima informazione in entrata sia diverso da zero si crea il primo elemento di lista, si assegna `x.inf` e si inizializza il puntatore ausiliario `paus` al valore `p` (`punt_lista`):

```
p = (struct elemento *)malloc(sizeof(struct elemento));
p->inf = x.inf;
paus = p;
```

Inizia poi il ciclo di ingresso delle altre informazioni. Se il valore dell'informazione è diverso da zero si procede alla creazione di un nuovo elemento di lista e all'inserimento dell'informazione nel corrispondente campo:

```
/* creazione dell'elemento successivo */
paus->pun = (struct elemento *)malloc(sizeof(struct elemento));

paus = paus->pun;          /* Attualizzazione di paus */
paus->inf = x.inf;         /* Inserimento dell'informazione
                           nell'elemento */
```

La soluzione è simile a quella vista per la visualizzazione. Si tratta di scandire la lista per cercare il valore maggiore. Il `main` passa alla funzione il puntatore alla lista `punt_lista` e il valore di ritorno della funzione è il maggiore che viene visualizzato:

```
printf("\nIl maggiore è: %d", maggiore_lista(punt_lista));
```

All'interno di `maggiore_lista` dichiariamo la variabile `max` di tipo `int`, dove inseriamo e successivamente visualizziamo il maggiore fra gli elementi; `max` deve essere inizializzato a `MIN_INT`, costante definita in `limits.h`, contenente il minimo intero rappresentabile.

Si scorre la lista lineare nello stesso modo della funzione `visualizza_lista`, ma a ogni ciclo si controlla se il campo informazione dell'elemento considerato sia maggiore di `max`, nel qual caso si effettua l'assegnamento:

```
max = p->inf;
```

```
/* Accetta in ingresso una sequenza di interi e li
```

```

    memorizza in una lista. La sequenza termina quando
    viene immesso il valore zero. La lista creata viene
    visualizzata. Determina il maggiore della lista    */

#include <stdio.h>
#include <malloc.h>
#include <limits.h>

struct elemento {
    int inf;
    struct elemento *pun;
};

struct elemento *crea_lista2();
void visualizza_lista(struct elemento *);
int maggiore_lista(struct elemento *);

main()
{
    struct elemento *punt_lista; /* Puntatore alla testa
                                   della lista          */
    punt_lista = crea_lista2(); /* Chiamata funzione per
                                   creare la lista        */
    visualizza_lista(punt_lista); /* Chiamata funzione per
                                   visualizzare la lista */
    /* Stampa il valore di ritorno della funzione maggiore_lista() */
    printf("\nIl maggiore e': %d\n\n", maggiore_lista(punt_lista));
}

/* Accetta in ingresso una sequenza di interi e li
   memorizza in una lista. Il numero di interi che compongono
   la sequenza termina con il valore zero */
struct elemento *crea_lista2()
{
    struct elemento *p, *paus;
    struct elemento x;

    printf("\nInserisci un'informazione (0 per fine lista): ");
    scanf("%d", &x.inf);

    if(x.inf==0) p = NULL; /* Lista vuota */
    else {
        /* Creazione del primo elemento */
        p = (struct elemento *)malloc(sizeof(struct elemento));

        p->inf = x.inf;
        paus=p;

        while(x.inf!=0) {
            printf("\nInserisci un'informazione (0 per fine lista): ");
            scanf("%d", &x.inf);
            if(x.inf!=0) {
                /* creazione dell'elemento successivo */
                paus->pun = (struct elemento *)malloc(sizeof(struct elemento));

                paus = paus->pun; /* Attualizzazione di paus */
                paus->inf = x.inf; /* Inserimento dell'informazione
                                   nell'elemento          */
            }
        }
    }
}

```

```

    }
    else
        paus->pun = NULL;        /* Marca di fine lista */
    }
}
return(p);
}

/* Determina il maggiore della lista.
   Il parametro in ingresso è il puntatore alla testa */
maggiore_lista(struct elemento *p)
{
    int max = INT_MIN;

    /* Ciclo di scansione della lista */
    while(p != NULL) {
        if(p->inf > max)
            max = p->inf;
        p = p->pun;                /* Scorre di un elemento in avanti */
    }
    return(max);
}

/* Visualizza la lista */
void visualizza_lista(struct elemento *p)
{
    printf("\npunt_lista---> ");

    /* Ciclo di scansione della lista */
    while(p!=NULL) {
        printf("%d", p->inf);      /* Visualizza il campo informazione */
        printf("---> ");
        p = p->pun;                /* Scorre di un elemento in avanti */
    }
    printf("NULL\n\n");
}

```

Listato 14.2 Programma che crea, visualizza la lista e ne determina il maggiore. L'immissione dei valori da parte dell'utente termina con zero