





Facoltà di Scienze
Matematiche
Fisiche Naturali

Laboratorio di Algoritmi e Strutture Dati

Prof. Aniello Murano

Implementazioni di Liste Doppiaemente Puntate e Circolari

Corso di Laurea
Codice insegnamento
Email docente
Anno accademico

Informatica
13917
murano@na.infn.it
2007/2008

Lezione numero: 9

Parole chiave: Liste dinamiche, liste circolari, liste doppiamente puntate








13/11/2008



Facoltà di Scienze
Matematiche
Fisiche Naturali

Indice

- **Liste puntate semplici:** Gli elementi sono organizzati in modo sequenziale e si possono scorrere in un unico verso. La lista ha un primo elemento (testa) e un ultimo elemento (coda)


- **Liste doppiamente puntate:** Sono simili alle liste puntate semplici, ma permettono di scorrere gli elementi in entrambi i versi


- **Liste puntate semplici circolari:** Sono liste puntate semplici senza testa ne coda.


- **Liste doppiamente puntate circolari:** Liste doppiamente puntate senza testa ne coda.









Federica 13/11/2008 3

Facoltà di Scienze
Matematiche
Fisiche Naturali

Liste puntate doppie

Una Lista Doppia Puntata è un insieme dinamico in cui ogni elemento ha uno o più campi contenenti informazioni e due riferimenti, uno all'elemento successivo (next) della lista ed uno all'elemento precedente (prev) della lista.

back X next

Federica 13/11/2008 4

Facoltà di Scienze
Matematiche
Fisiche Naturali

Implementazione in C

- Per definire la struttura di un elemento di una lista doppiamente puntata bisogna utilizzare due puntatori alla stessa struttura. Il primo punterà all'elemento precedente mentre il secondo punterà a quello successivo:


```
struct el
{
    struct el *prev;
    int inf;
    struct el *next;
}
```
- Per l'inizializzazione di una lista doppiamente puntata si può considerare il codice visto per l'inizializzazione delle liste singolarmente puntate, introducendo opportunamente il codice per la gestione del puntatore all'elemento precedente.

back X next



Federica
UNIVERSITÀ

13/11/2008



5
Facoltà di Scienze
Matematiche
Fisiche Naturali

Funzione crea_lista() 1/2

- La funzione **crea_lista()** crea due puntatori ad elemento, uno di nome **p** (puntatore al primo elemento della lista) e l'altro di nome **punt** (puntatore che permette di scorrere la lista);

```

struct el *crea_lista() {
    struct el *p, *punt;
    int i, n;
    printf("\n Specificare il numero di elementi... ");
    scanf("%d", &n);
    if(n==0)
        p = NULL;
    else {
        /* creazione primo elemento */
        p = (struct el *)malloc(sizeof(struct el));
        printf("\n Inserisci il primo valore: ");
        scanf("%d", &p->inf);
        punt = p; p->prev=NULL;
    }
}

```

→

→

→

prev	inf	next
p →	5	
	↑	
	punt	

back
✖
next



Federica
UNIVERSITÀ

13/11/2008



6
Facoltà di Scienze
Matematiche
Fisiche Naturali

Funzione crea_lista() 2/2

```

for(i=2; i<=n; i++)
{
    punt->next = (struct el *)malloc(sizeof(struct el));
    punt->next->prev=punt;
    → punt = punt->next;
    printf("\n Inserisci il %d elemento: ", i);
    scanf("%d", &punt->inf);
} // chiudo il for
punt->next = NULL; // marcatore fine lista
} // chiudo l'if-else
return(p);
} // chiudo la funzione

```

back
✖
next

Federica 13/11/2008 7  Facoltà di Scienze
Matematiche
Fisiche Naturali

Inserimento in coda: main()

```
#include <stdio.h>
#include <malloc.h>
struct el {struct el *prev; int inf; struct el *next;};
struct el *crealista();
struct el *inserisci(struct el *,int);
int main() {
    struct el *lista;
    int valore;
    lista=crealista(); stampalista();
    printf("\nInserisci elemento da inserire: ");
    scanf("%d", &valore);
    lista=inserisci (lista,valore);
}
```

back  next

Federica 13/11/2008 8  Facoltà di Scienze
Matematiche
Fisiche Naturali

Inserimento in coda: funzione

```
struct el *inserisci(struct el *p, int valore)
{
    struct el *nuovo=NULL,
    struct el *testa;
    if (p==NULL) {
        p=(struct el *)malloc(sizeof(struct el));
        p->inf=valore; p->prev=NULL; testa=p; }
    else {
        testa=p;
        while (p->next!= NULL)
            p=p->next;
        nuovo=(struct el *)malloc(sizeof(struct el));
        nuovo->prev=p;
        nuovo->inf=valore;
        nuovo->next = NULL; }
    p->next=nuovo;
    return testa;
}
```

back  next

Federica 13/11/2008 9 Facoltà di Scienze Matematiche Fisiche Naturali

Liste puntate circolari

- Una Lista Circolare puntata è un una lista puntata in cui il puntatore next dell'ultimo elemento della lista punta all'elemento in testa alla lista.
- Infine, se la lista è doppiamente puntata, il puntatore prev della testa della lista punta all'elemento in coda alla lista

back X next

Federica 13/11/2008 10 Facoltà di Scienze Matematiche Fisiche Naturali

Funzione crea_lista() 1/2

- La funzione **crea_lista()** crea due puntatori ad elemento, uno di nome **p** (al primo elemento della lista) e l'altro di nome **punt** (che permette di scorrere la lista);

```

struct el *crea_lista() {
    struct el *p, *punt;
    int i, n;
    printf("\n Specificare il numero di elementi... ");
    scanf("%d", &n);
    if(n==0) p = NULL;
    else {
        /* creazione primo elemento */
        p = (struct el *)malloc(sizeof(struct el));
        printf("\nInserisci il primo valore: ");
        scanf("%d", &p->inf);
        punt = p; p->prev=NULL;
    }
}

```

back X next

Federica 13/11/2008 11 Facoltà di Scienze Matematiche Fisiche Naturali

Funzione crea_lista() 2/2

```

for(i=2; i<=n; i++)
{
    punt->next = (struct el *)malloc(sizeof(struct el));
    punt->next->prev=punt;
    punt = punt->next;
    printf("\nInserisci il %d elemento: ", i);
    scanf("%d", &punt->inf);
} // chiudo il for
→ punt->next = p; p->prev = punt; // circolarità della lista
} // chiudo l'if-else
return(p);
} // chiudo la funzione

```

back X next

Federica 13/11/2008 12 Facoltà di Scienze Matematiche Fisiche Naturali

Inserimento all'interno di liste doppiamente puntate

- Supponiamo di voler inserire un elemento con valore **val** in una lista prima dell'elemento puntato da **pos**

```

struct el *ins_in_testa(stuct el *pos, int val)
{
    LISTA *p;
    p = (struct el *)malloc(sizeof(struct el));
    p -> inf = val;
    /* aggiornamento dei puntatori */
    p -> next = pos;
    p -> prev=NULL;
    if !(pos==NULL) {
        p -> prev = pos -> prev;
        pos->prev=p;
        if !(p -> prev== NULL)
            p->prev->next=p;
    }
    return (p);
}

```

testa coda

back X next

Federica 13/11/2008 13 Facoltà di Scienze Matematiche Fisiche Naturali

Eliminazione elemento da una lista circolare doppiamente puntata

```

struct el *togli(struct el *p, int valore)
{
    struct el *inizio=NULL; int i=1;
    if (p!=NULL)
    {
        inizio=p;
        do
        {
            if (p->inf==valore)
            {
                if (p->next==p) inizio=NULL;
                else {
                    p->prev->next=p->next;
                    p->next->prev=p->prev;
                    if p==inizio inizio=p->next; }
                free(p);
                p=NULL;
            }
            else p=p->next;
        } while (p!=inizio && p!=NULL);
    }
    return inizio;
}

```

Punterà alla testa della lista

Nel caso in cui il nodo rimosso è la testa della lista

Vero se il valore cercato è nella lista

Controlla che tutta la lista è stata letta

back X next

Federica 13/11/2008 14 Facoltà di Scienze Matematiche Fisiche Naturali

Esercizio 1

Siano L1 e L2 due liste non circolare doppiamente puntate definite da

```

struct el
{
    struct el *prev;
    int inf;
    struct el *next;
}L1, L2;

```

Si supponga che L1 e L2 siano della stessa lunghezza. Scrivere una funzione ricorsiva in linguaggio C che prese in input le due liste L1 e L2, cambi L2 inserendo (prima o dopo di) ogni elemento in L2 un elemento in L1.

Scrivere poi una funzione ricorsiva che rimuova da L2 tutti i valori negativi.

back X next

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.