

# OPERAZIONI SULLE LISTE

venerdì 10 novembre 2017 09:10

→ ABBIAMO GIÀ VISTO INSERIMENTO IN TESTA E IN CODA

→ CHE COSA ABBIAMO IMPARATO:

→ SE LA PROCEDURA DEVE POTER MODIFICARE IL PRIMO ELEMENTO LA LISTA DEVE ESSERE PASSATA PER INDIRIZZO (PUNTIATRE)

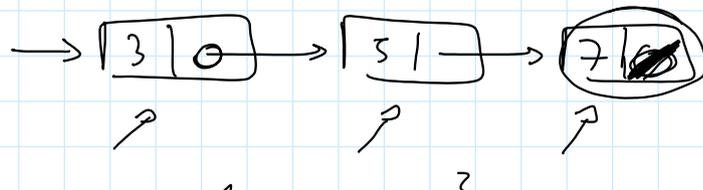
→ SE DOBBIAMO SCORRERE LA LISTA LO POSSIAMO FARE USANDO UN PUNTATORE TEMPORANEO (ES. COLL) CHE SI AGGIORNA ALL'INTERNO DI UN CICLO

OPERAZIONE: Scrivere una funzione che, data una lista, ne calcola la lunghezza

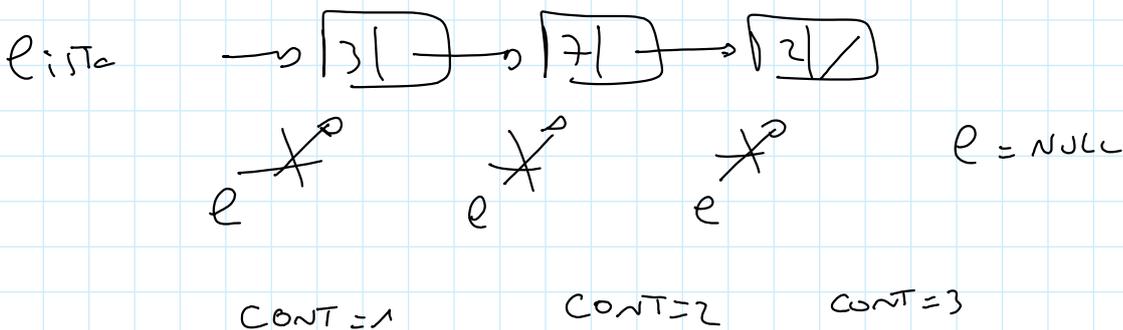
```

int length (ListaDiElementi e) {
    int cont = 0;
    while (e != NULL) {
        cont++;
        e = e -> next;
    }
    return cont;
}
    
```

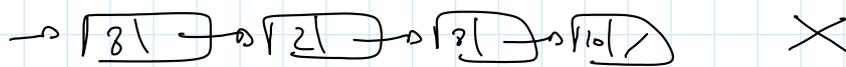
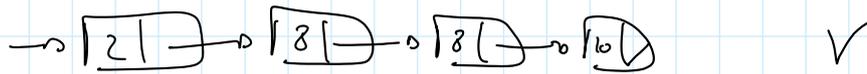
NOT E' NECESSARIO  
 PASSARE LA LISTA  
 PER MODIFICARLA  
 PERCHE' NON MODIFICO IL PRIMO ELEMENTO



~~con e~~      con e      e  
 POSSO USARE e PER SCORRERE  
 LA LISTA PERCHE' E' UNA COPIA  
 DEL PUNTATORE AL PRIMO ELEMENTO



OPERAZIONE: Scrivere una funzione che verifica se una lista data è ordinata in modo non decrescente (ogni elemento  $e_i \geq e_{i-1}$ )



```
int ord (ListaDiElementi e) {
```

```
    int ordinata = 1;
```

```
    while (e->next != NULL && ordinata)
```

```
    while (e != NULL && e->next != NULL && ordinata) {
```

```
        if (e->info > e->next->info)
```

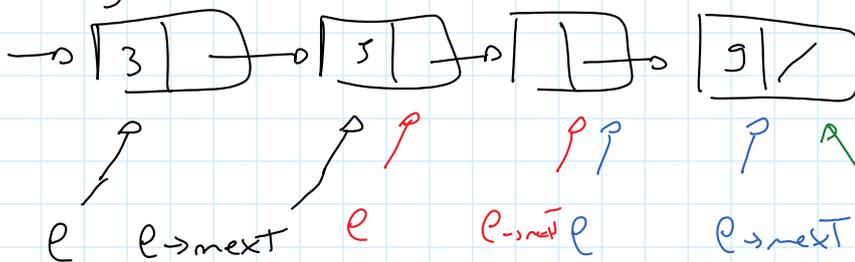
```
            ordinata = 0;
```

```
        else e = e->next;
```

```
    } return ordinata;
```

ANCHE IN QUESTO CASO POSSIAMO PASSARE LA LISTA PER VALORE

PROBLEMA: SE LA LISTA È VUOTA E → NEXT DA ERRORE!!  
 AGGIUNGO UNA CONDIZIONE CHE NON MI FA ESEGUIRE IL CICLO IN CASO DI LISTA VUOTA



e → next  
 NULL

IN QUESTA SITUAZIONE HO TERMINATO L'OPERAZIONE

$e \rightarrow next \rightarrow info$

accede al valore "info" dell'elemento  
successivo a quello puntato da  $e$

nel while è importante scrivere

$e \neq \text{NULL} \ \&\& \ e \rightarrow \text{next} \neq \text{NULL} \ \&\& \dots$

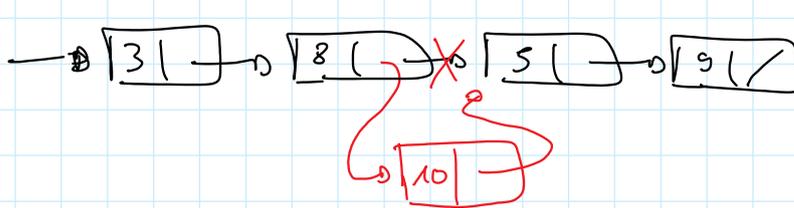
IN QUEST'ORDINE PERCHÉ LA PRIMA  
CONDIZIONE ASSICURA DI NON AVERE  
ERRORE CON  $e \rightarrow \text{next}$  QUANDO  $e = \text{NULL}$

# INSERIMENTO DI UN ELEMENTO "NEL MEZZO"

venerdì 10 novembre 2017 09:49

## DI UNA LISTA

Scrivere una procedura che inserisce un <sup>nuovo</sup> elemento con valore  $v$  subito prima della prima occorrenza di un elemento con valore  $x$ .  
Se  $x$  non è presente e esiste rimane inalterata.



$v = 10$

$x = 5$

È POSSIBILE CHE  
LA PROCEDURA MODIFICH  
LA TESTA DELLA  
LISTA

```
void (ListaDiElementi *l, int x, int v) {
```

```
    ListaDiElementi* prec, conn;
```

```
    prec = NULL;
```

```
    conn = *l;
```

```
    int Trovato = 0;
```

```
    while (conn != NULL && !trovato) {
```

```
        if (conn->info == x)
```

```
            trovato = 1;
```

```
        else {
```

```
            prec = conn;
```

```
            conn = conn->next;
```

```
        }
```

```
    }
```

```
    if (!trovato) {
```

SE  $x$  È STATO TROVATO  
... ACCI...

```

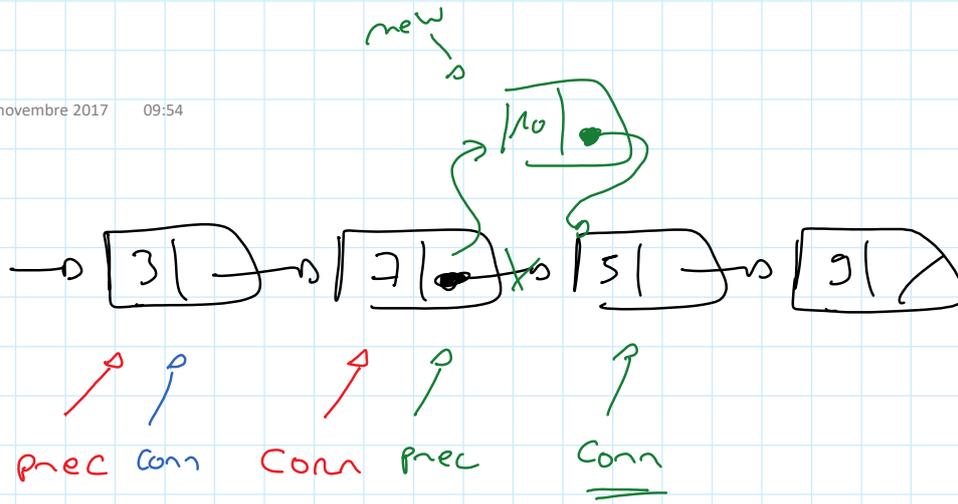
)
if (Trovato) {
    listaDiElementi new =
        malloc(sizeof(elemento della lista));
    new -> info = v;
    new -> next = con;
    if (prec != NULL)
        prec -> next = new;
    else
        *p = new;
}
}

```

SE X È STATO TROVATO  
 DEVO AGGIUNGERE IL  
 NUOVO ELEMENTO  
 ALTRIMENTI NON  
 DEVO FARE  
 NULLA

VEDO SE SONO  
 NEL MEZZO  
 DELLA LISTA

DEVO INTERARE  
 IN TESTA  
 (MODIFICO IL  
 PUNTIATORE  
 ALL'INIZIO  
 DELLA LISTA)



prec = NULL

## VARIANTI DI QUESTO ESEMPIO (PER ESERCIZIO)

→ invece di  $e_c$  lasciare  $e_c$  e sta immutata  
inserire  $v$  in coda se  $x$  non è  
presente

→ inserire  $v$  dopo la prima occorrenza di  $x$

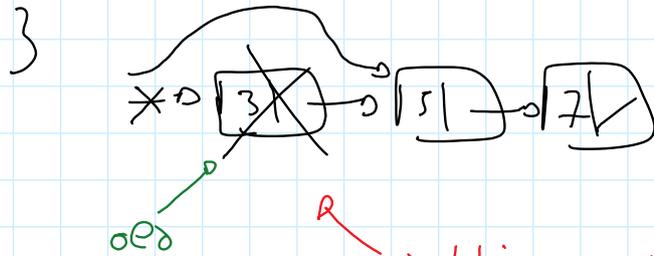
→ inserire  $v$  dopo l'ultima occorrenza di  $x$

# ELIMINAZIONE DELL'ELEMENTO IN TESTA

venerdì 10 novembre 2017 10:27

ALLA LISTA (se la lista non è vuota)

```
void cancella (ListaDiElementi *l) {  
    if (*l != NULL) {  
        ListaDiElementi old = *l;  
        *l = *l->next;  
        free (old);  
    }  
}
```

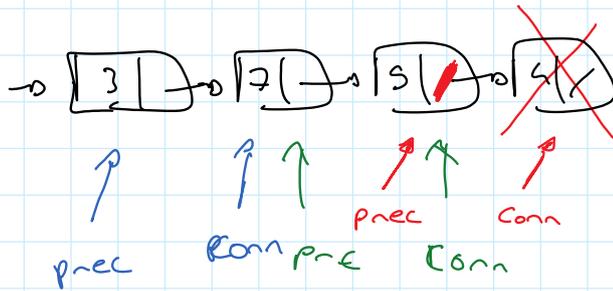


dobbiamo usare la free(...) per liberare la memoria nello heap

LA FREE SI ASPETTA UN PUNTIATORE CHE POI NON SARÀ PIÙ UTILIZZABILE

# ELIMINAZIONE DELL'ELEMENTO IN CODA ALLA LISTA

venerdì 10 novembre 2017 10:34



ATTENZIONE:  
SE LA LISTA CONTIENE  
UN SOLO ELEMENTO  
DEVO MODIFICARE IL  
PUNTIATORE ALL'INIZIO  
DELLA LISTA

```
void cancella (ListaDiElementi *l) {
```

```
    if (*l != NULL) {
```

```
        if ((*l) -> next == NULL) {
```

CASO  
LISTA CON  
UN SOLO  
ELEMENTO

```
            free(*l);
```

```
            *l = NULL;
```

DEVO FARE  
PRIMA LA  
FREE ALTRIMENTI  
DOPO NON POTREI  
PIU' FARLA...

```
        } else {
```

CASO  
PIU' DI  
UN ELEMENTO

```
            ListaDiElementi prec = *l;
```

```
            ListaDiElementi conn = (*l) -> next;
```

```
            while (conn -> next != NULL) {
```

```
                prec = conn;
```

```
                conn = conn -> next;
```

$(*l) \rightarrow next$   
ci vogliamo  
le parentesi  
 $l_c \rightarrow l_c$   
 $l_c$  precedente  
su \*

```
}
```

```
free (curr);
```

```
prec → next = NULL ;
```

```
}
```

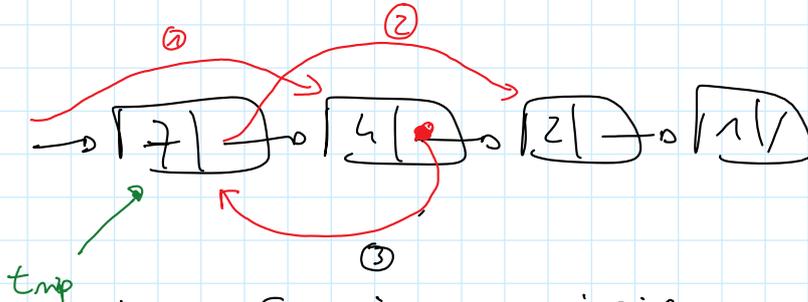
```
}
```

```
}
```

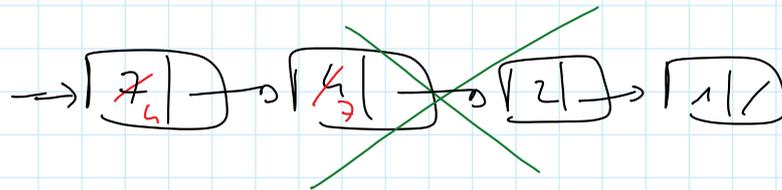
# SPOSTAMENTO DI ELEMENTI

venerdì 10 novembre 2017 10:50

scrivere una procedura che scambia i primi due elementi di una lista



Lo faccio modificando i puntatori, non i campi info dei vari elementi



```
void scambiaprimidue (ListaDiElementi *l) {
```

```
if (*l != NULL && (*l) -> next != NULL) {
```

```
ListaDiElementi tmp = *l;
```

```
*l = (*l) -> next;
```

```
tmp -> next = (*l) -> next;
```

```
(*l) -> next = tmp;
```

```
}
```

```
}
```

\*l != NULL  
mi ASSICURO  
CHE  
(\*l) -> next  
NON DA  
ERRORE

## ESERCIZI PER CASA:

Scrivere opportune procedure o funzioni che eseguano le seguenti operazioni:

- 1) Eliminare un elemento che precede la prima occorrenza di un dato valore  $x$

VARIANTI : - "SEGUE" al posto di "PRECEDE"  
- "ULTIMA OCCORRENZA" al posto di "PRIMA OCCORRENZA"

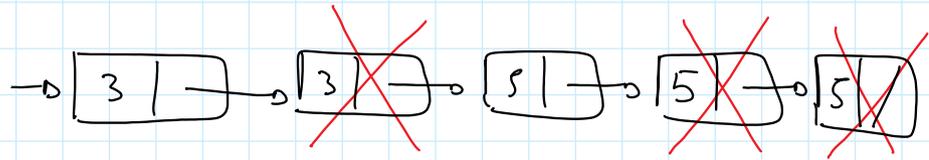
- 2) Restituire il valore massimo contenuto nella lista. Se la lista è vuota restituire 0. La lista può contenere anche valori negativi.

- 3) Scambiare nella lista l'elemento massimo e quello minimo (la prima occorrenza di ognuno dei due)

- 4) Eliminare Tutti gli elementi della lista

- 5) Eliminare Tutti gli elementi immediatamente successivi ad un elemento con lo stesso valore. Ad esempio:

valore. Ad esempio:



PER ALTRI ESERCIZI VEDERE I TESTI D'ESAME  
DEGLI ANNI PASSATI