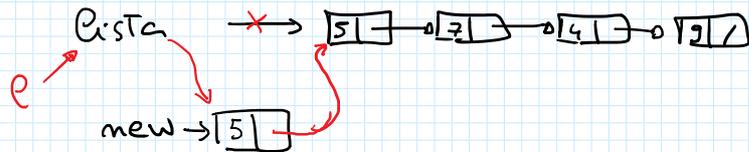


# OPERAZIONI SULLE LISTE

martedì 8 novembre 2016 08:53

```
struct el { int info; struct el * next; }  
typedef struct el ElementoDiLista;  
typedef ElementoDiLista * ListaDiElementi
```

ABBIAMO VISTO: Aggiungere un elemento "in Testa" ad una lista



```
void addT (ListaDiElementi *e, int x)  
{  
    ListaDiElementi new = malloc(sizeof(ElementoDiLista));  
    new -> info = x;  
    new -> next = *e;  
    *e = new;  
}
```

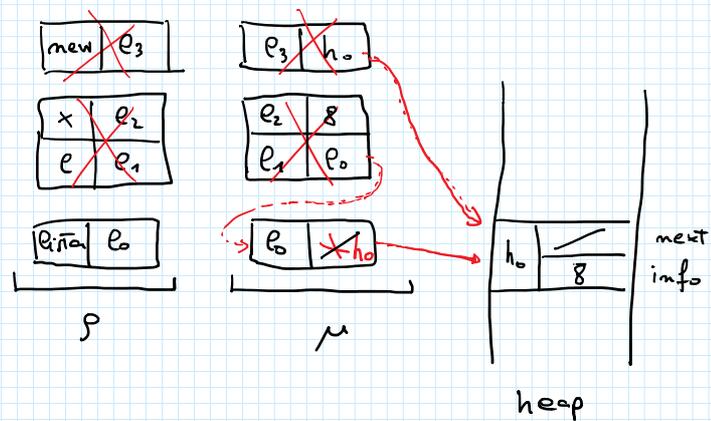
CHIAMATA  
addT (&Lista, 5);

Una lista vuota (contenente  $\emptyset$  elementi) è rappresentata da un puntatore a NULL di Tipo ListaDiElementi

```
main() {
  :
  ListaDiElementi lista = NULL;
  addT (&lista, 8);
  :
}
```

FUNZIONA SU UNA  
LISTA VUOTA?

SI!



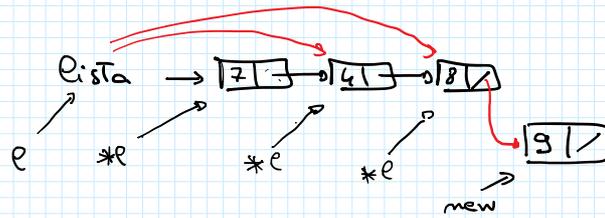
- addT può essere utilizzata per riempire una lista di elementi
- i valori dovranno essere passati ad addT in ordine inverso rispetto a come vogliamo che compaiano nella lista

```
main ()  
{  
    listaDiElementi lista = NULL;  
    addT (&lista, 5);  
    addT (&lista, 8);  
    addT (&lista, 4);  
}
```

#### RISULTATO

lista → [4] → [8] → [5]

## AGGIUNGERE UN ELEMENTO "IN CODA" AD UNA LISTA



```
void addC ( ListaDiElementi * l, int x )
```

```
{
```

```
  ListaDiElementi new = malloc (sizeof (ElementoDiLista));
```

```
  new → info = x;
```

```
  new → next = NULL;
```

```
  if (*l == NULL) *l = new;
```

```
  else
```

```
  {
```

```
    while (*l → next != NULL)
```

```
      *l = *l → next;
```

```
    *l → next = new;
```

```
  }
```

```
}
```

SBAGLIATO !!

MODIFICARE \*l CORRISPONDE  
A MODIFICARE lista

⇒ HO DISTRUITO LA LISTA  
CHE MI È STATA PASSATA

SOLUZIONE CORRETTA: LAVORARE SU UNA COPIA DI \*l

```
void addC (ListaDiElementi *l, int x)
```

```
{
```

```
  ListaDiElementi new = malloc (sizeof (ElementoDiLista));
```

```
  new → info = x;
```

```
  new → next = NULL;
```

```
  if (*l == NULL) *l = new;
```

```
  else
```

```
  {
```

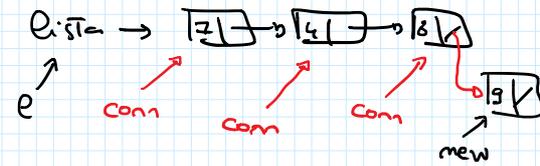
```
    ListaDiElementi curr = *l;
```

```
    while (curr → next != NULL)
```

```
      curr = curr → next;
```

```
    curr → next = new;
```

```
  }
```



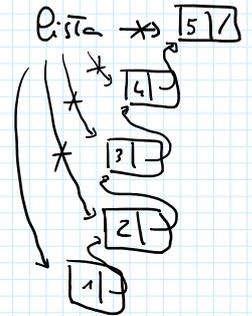
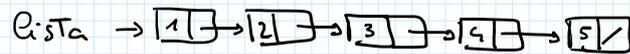
ESEMPIO : costruire una lista contenente i primi  $n$  numeri interi

```
main()
{
  int m=5;
  ListaDiElementi lista = NULL;

  for (int i=1; i<=m; i++)
    addC(&lista, i);
}
```

ALTERNATIVA

```
for (int i=m; i>0; i--)
  addT(&lista, i);
```



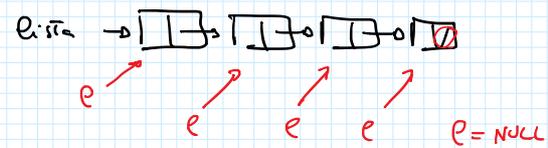
ENTRAMBE LE SOLUZIONI SONO  
CORRETTE. QUELLA CHE USA addT È  
PIÙ EFFICIENTE IN QUANTO EVITA DI  
SCANDIRE TUTTE LE VOLTE L'INTERA LISTA

ESEMPIO: calcolo della lunghezza di una lista

```

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

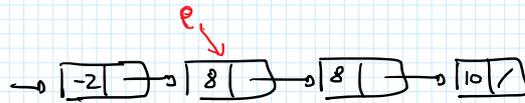
```



Lunghezza ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4 =

Chiamata  
length(Lista)

ESEMPIO : Data una lista controllare se è ordinata in modo non decrescente



✓ True



✗ False

int ond (Lista Di Elementi e)

```

{
  int ordinata = 1;
  while (e != NULL && ordinata) while (e != NULL && e->next != NULL
  {
    if ((e->info) > (e->next->info))
      ordinata = 0;
    else
      e = e->next;
  }
  return ordinata;
}

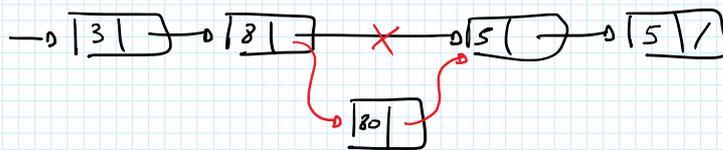
```

RESTISCE IL CASO LISTA VUOTA  
 FA FERMARE IL CICLO AL PENULTIMO ELEMENTO

L'ORDINE DI QUESTE DUE CONDIZIONI È IMPORTANTE NEL CASO e = NULL

## INSERIMENTO DI UN ELEMENTO "NEL MEZZO" DI UNA LISTA

Scrivere una procedura che inserisce in una lista un valore  $V$  subito prima di un dato valore  $x$ . Se  $x$  non è presente nella lista, la lista deve rimanere inalterata.

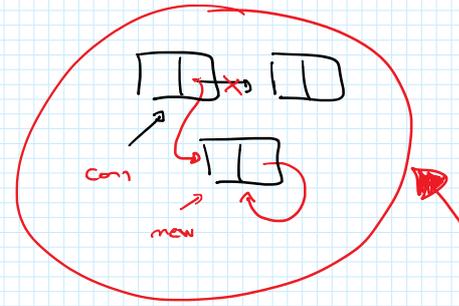
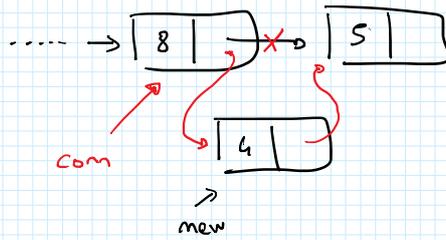

 $V = 80$ 
 $x = 5$ 

 $V = 80$ 
 $x = 100$ 

LISTA INALTERATA!

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

IDEA:



- usiamo un puntatore `com` per scandire la lista
- fermiamo la scansione quando `com` punta all'elemento che precede `x`

```

void insprima (ListaDiElementi *l, int x, int v)
{
    ListaDiElementi com = *l;
    int trovato = 0;
    while (com != NULL && com->next != NULL && !trovato)
    {
        if (com->next->info == x)
            trovato = 1;
        else
            com = com->next;
    }
    if (trovato)
    {
        ListaDiElementi new = malloc(sizeof(ElementoDiLista));
        new->info = v;
        new->next = com->next;
        com->next = new;
    }
}

```

VANO FATTE IN  
QUESTO  
ORDINE ... ALTRIMENTI

VARIANTI PER ESERCIZIO :

- INSERIRE `v` IN FONDO ALLA LISTA SE `x` NON È PRESENTE
- INSERIRE `v` SUBITO DOPO LA PRIMA OCCORRENZA DI `x` NELLA LISTA
- INSERIRE `v` SUBITO DOPO L'ULTIMA OCCORRENZA DI `x` NELLA LISTA