Liste

```
struct el
{ int info;
  struct el* next;
}
```

```
typedef struct el ElementoDiLista;
```

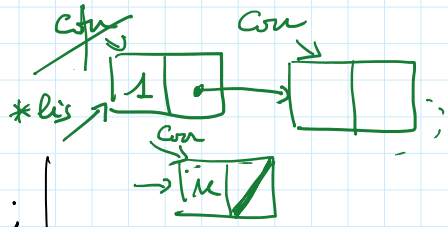
tipo — nuovo nome

```
typedef ElementoDiLista * ListaDiElementi;
```

tipo — nuovo nome

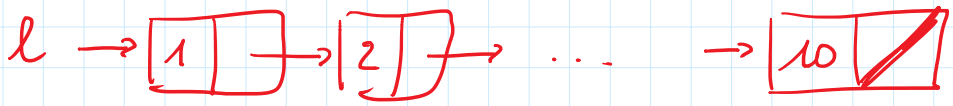
Abbiamo visto una procedura per creare una lista di n elementi con valori 1, 2, ..., n

```
void creaLista (ListaDiElementi * lis, int n)
{ int i;
  ListaDiElementi cor = malloc(sizeof(ElementoDiLista));
  * lis = cor;
  cor -> info = 1;
  for (i=2; i<=n; i++)
  { cor -> next = malloc( ... );
    cor = cor -> next;
    cor -> info = i;
  }
  cor -> next = NULL;
}
```

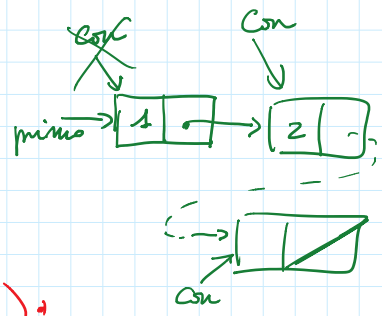


```
int main()
{   ListaDiElementi l;
    creaLista(&l, 10);
```

dopo questa chiamata l punta al primo elemento di una lista di 10 element



```
listeDiElementi creaLista (int n)
{   int i;
    listeDiElementi primo, corr;
    corr = malloc( ... );
    primo = corr;
    primo -> info = 1;
    for (i=2; i<=n; i++)
    {   corr -> next = malloc(...);
        corr = corr -> next;
        corr -> info = i;
    }
    corr -> next = NULL;
    return primo;
}
```

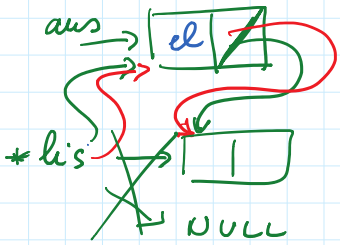


```
int main ()
{   ListaDiElementi l;
        l = crealista (10);
```

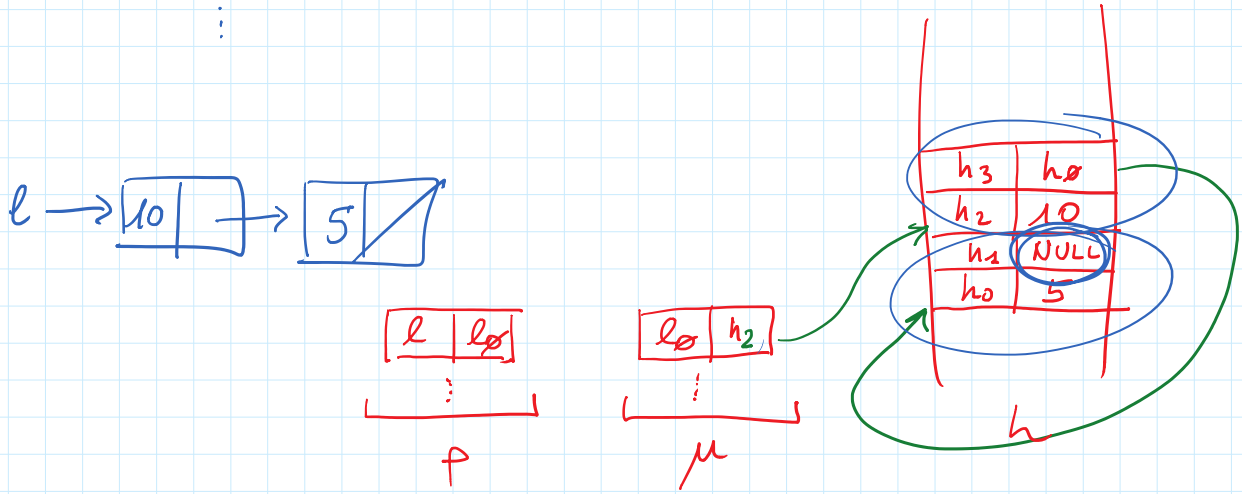
crealista crea la lista di 10 elementi e restituisce il puntatore al primo elemento. Il puntatore viene assegnato a l.

Procedura che aggiunge un elemento IN TESTA a una lista (possibilmente vuota).

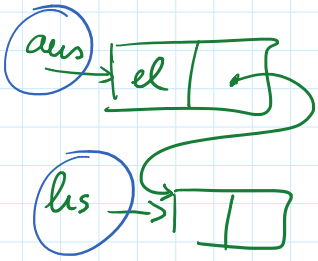
```
void addfirst (ListeDiElementi * lis, int el)
{
    ListeDiElementi aus = malloc( ... );
    aus -> info = el;
    aus -> next = *lis;
    *lis = aus;
}
```



```
int main()
{ ListeDiElementi l = NULL;
  addfirst(&l, 5);
  addfirst(&l, 10);
  ⋮
```

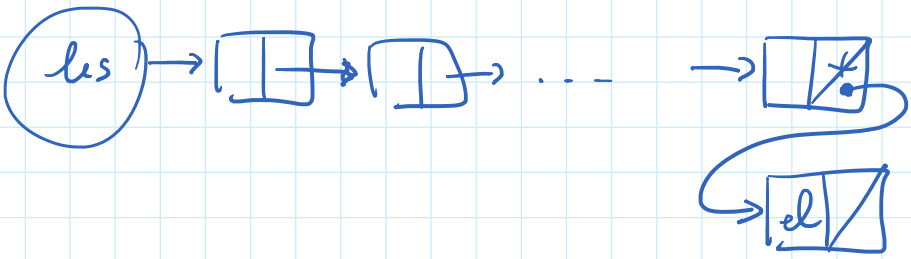


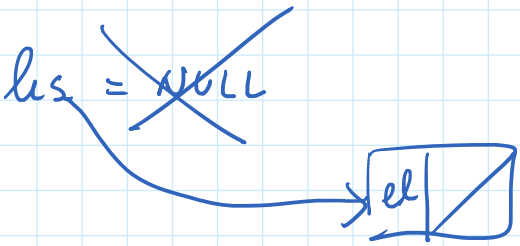
```
listaDiElementi addfirst (listaDiElementi lis, int el)
{ listaDiElementi aus = malloc(...);
  aus -> info = el;
  aus -> next = lis;
  return aus;
}
```



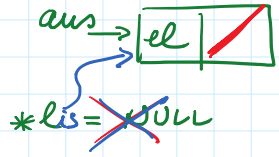

```
int main ()
{ listaDiElementi l = NULL;
  l = addfirst (l, 5);
  l = addfirst (l, 10);
```

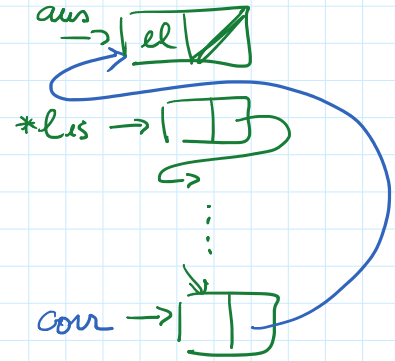
void addlast (ListeDiElement *lis, int el)





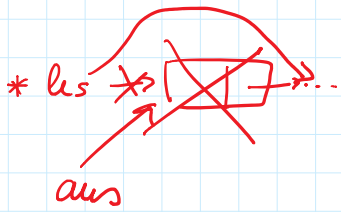
```
void addlast (ListeDiElementi * lis, int el)
{
    listeDiElementi aus = malloc( ... );
    aus -> info = el;
    aus -> next = NULL;
    if (* lis == NULL) * lis = aus;
    else
    {
        listeDiElementi corr = * lis;
        while (corr -> next != NULL)
            corr = corr -> next;
        corr -> next = aus;
    }
}
```





Cancellare da una lista il primo elemento (se c'è)

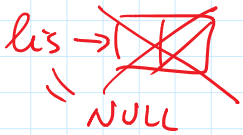
```
void delfirst (ListeDiElementi *lis)
{ if (*lis != NULL)
  { ListeDiElementi aus = *lis;
    *lis = *lis -> next;
    free (aus);
  }
}
```

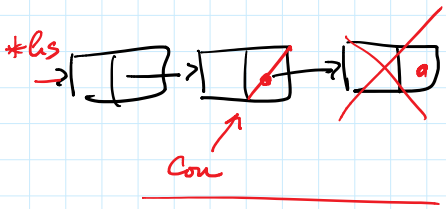


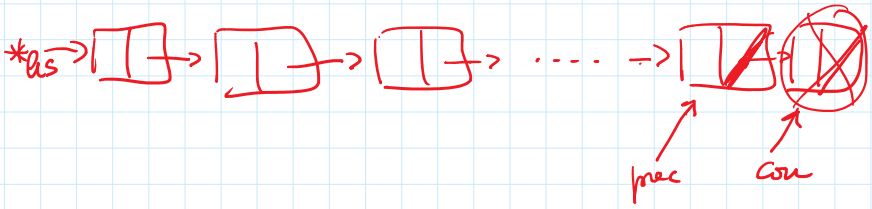
~~free (*lis)~~

Cancellare in una lista l'ultimo elemento (se c'è)

```
void dellast (ListeDiElementi * lis)
{ if (*lis != NULL)
    if (*lis -> next == NULL)
      { free (*lis);
        *lis = NULL;
      }
    else
      { ListeDiElementi cur = *lis;
        while (cur -> next -> next != NULL)
            cur = cur -> next;
        free (cur -> next)
        cur -> next = NULL;
      }
}
```

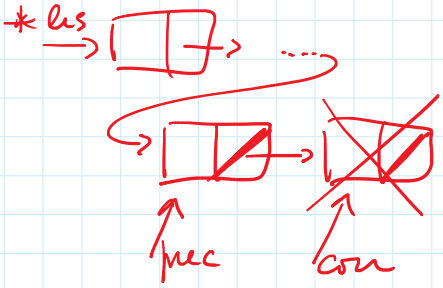




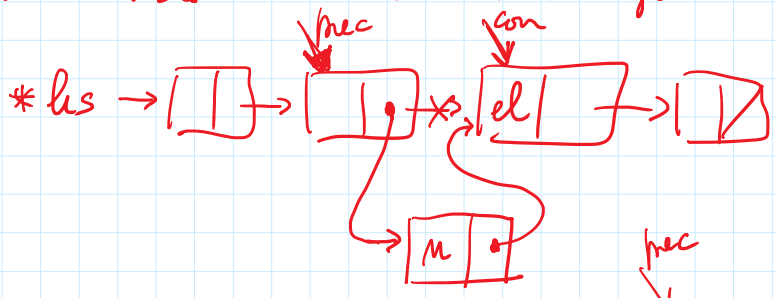


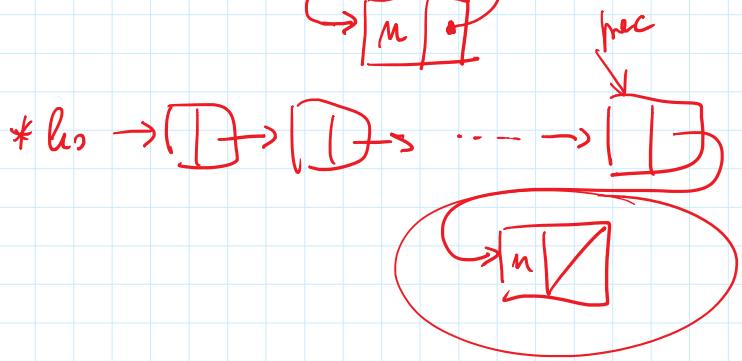
lunedì 7 novembre 2016 08:55

```
void delLast (ListeDiElementi * lis
{ if (*lis != NULL)
    if (*lis -> next == NULL)
      { free (*lis);
        *lis = NULL;
      }
    else
      { ListeDiElementi corr = *lis;
        ListeDiElementi prec = NULL;
        while (corr -> next != NULL)
          { prec = corr;
            corr = corr -> next;
          }
        free (corr);
        prec -> next = NULL;
      }
}
```

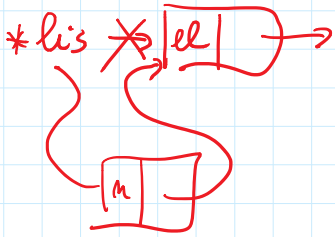


Procedura che inserisce un elemento con valore n prima della prima occorrenza di un valore el. Se il valore el non esiste si inserisce in fondo



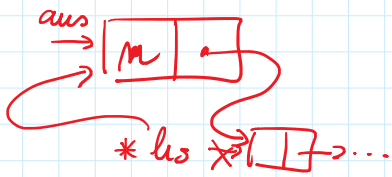


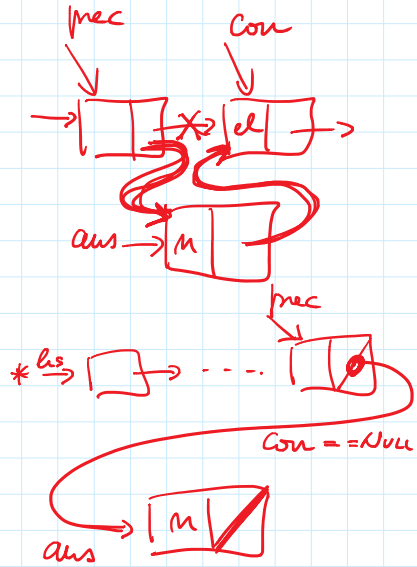
se el non esiste



```
void insbefore (listeDiElementi *lis, int el, int n)
{ int trovato = 0;
  listeDiElementi aus = malloc ( .... );
  listeDiElementi corr = *lis;
  listeDiElementi prec = NULL;
  aus -> info = n;

  while (corr != NULL && ! trovato)
     if (corr -> info == el) trovato = 1;
     else { prec = corr;
            corr = corr -> next;
          }
  if (trovato)
     if (prec == NULL)
       { aus -> next = *lis;
         *lis = aus;
       }
     else
       { aus -> next = corr;
         prec -> next = aus;
       }
  else
       { aus -> next = NULL;
         prec -> next = aus;
       }
}
```





16

lunedì 7 novembre 2016 08:55

17

lunedì 7 novembre 2016 08:55

18

lunedì 7 novembre 2016 08:55

19

lunedì 7 novembre 2016 08:55

20

lunedì 7 novembre 2016 08:55