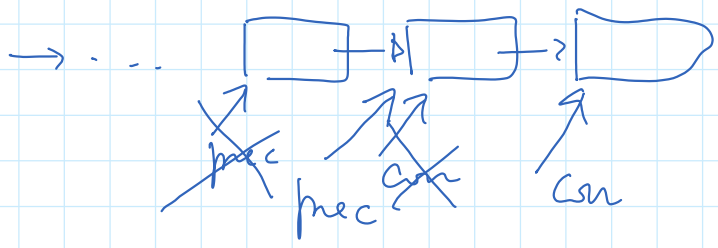
Inserire in una lista un elemento el prima della prima occorrenza di occ.
Se occ non esiste la lista rimane invariata.

```
void insbefore(int el, int occ,
               ElementoDiLista **l)
{ if (*l != NULL)
  { int trovato = 0;
    ElementoDiLista *corr = *l;
    ElementoDiLista *prec = NULL;
    while (corr != NULL && !trovato)
      if (corr->info == occ) trovato = 1;
      else { prec = corr;
             corr = corr->next;
           }
```

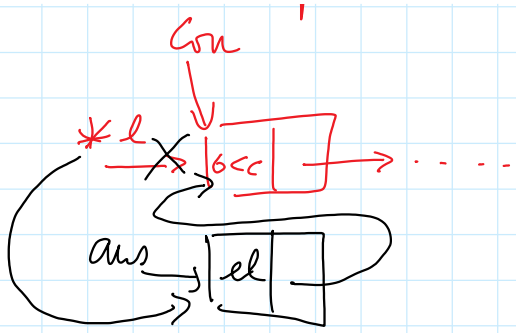



```
    if (trovato)
      if (prec == NULL)
```

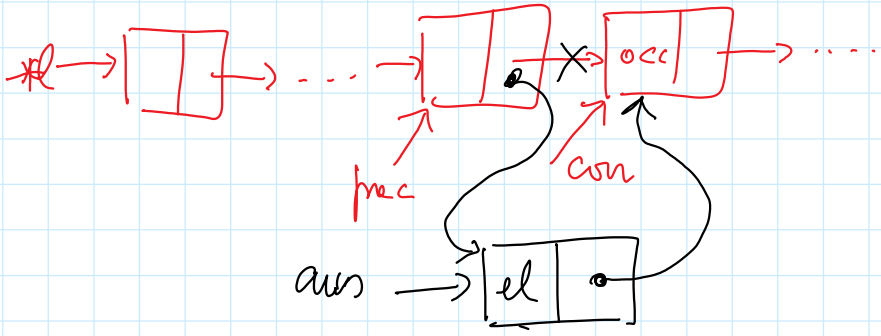
prec == NULL

corr

```
if (prec == NULL)
```

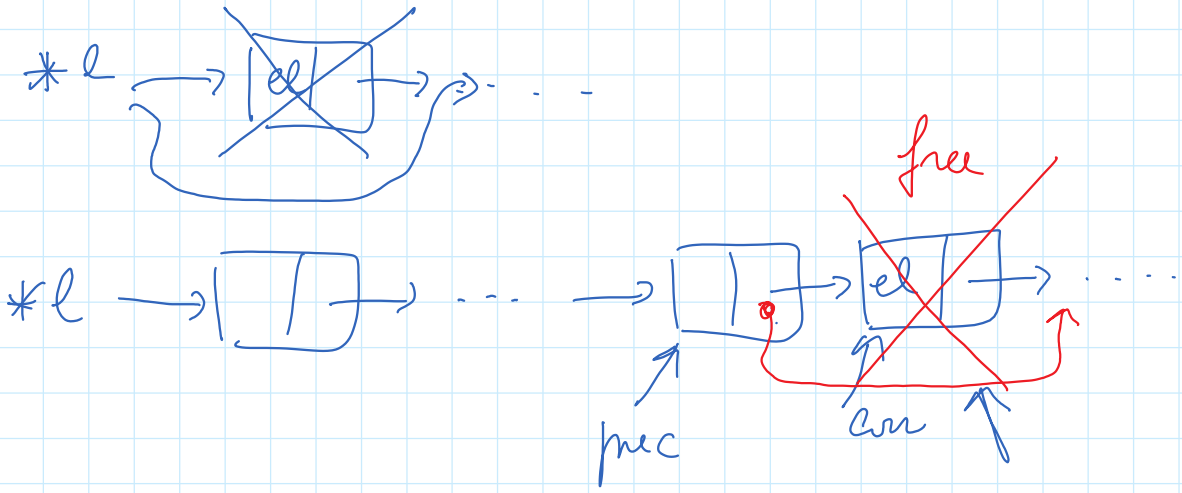



```
{ ElementoDiLista * aux = malloc(sizeof(EDL));
  aux -> next = *l;
  aux -> info = el;
  *l = aux;
}
else
{
```




```
ElementoDiLista * aux = malloc(sizeof(EDL));
aux -> next = Corr;
aux -> info = el;
prec -> next = aux;
}
}
}
```

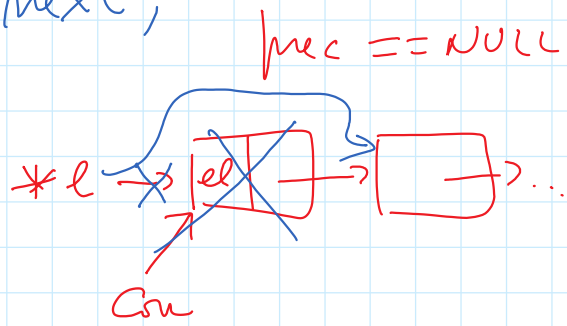
Scrivere una procedura che cancelli il primo elemento uguale a el. Se non esistono valori uguali a el le liste rimane invariate.



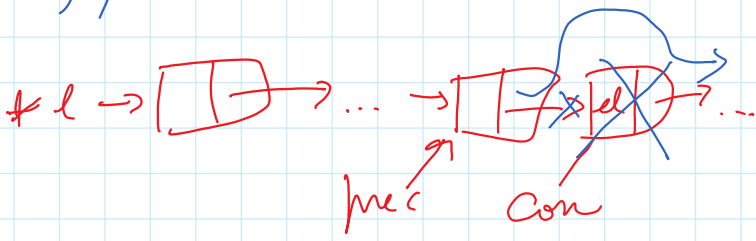

```
void canc (int el, ElementoDiLista **l)
{ if (*l != NULL)
  { int trovato = 0;
    ElementoDiLista * corr = *l;
    ElementoDiLista * prec = NULL;
    while (corr != NULL && !trovato)
      if (corr -> info == el) trovato = 1;
      else { prec = corr;
             corr = corr -> next;
           }
  }
}
```

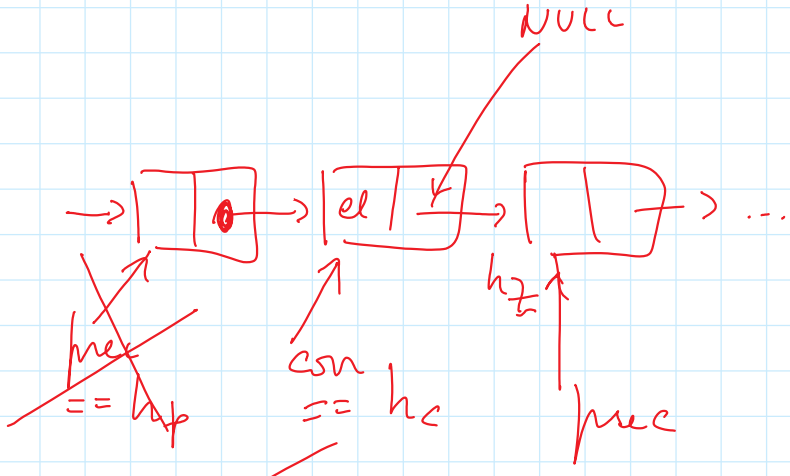
prec == NULL

```
        cur = cur -> next;
    }
if (trovato)
    if (prec == NULL)
    {
        *l = cur -> next;
        free (cur);
    }
else
{
    prec -> next = cur -> next;
    free (cur);
}
}
}
```



*l = (*l) -> next;

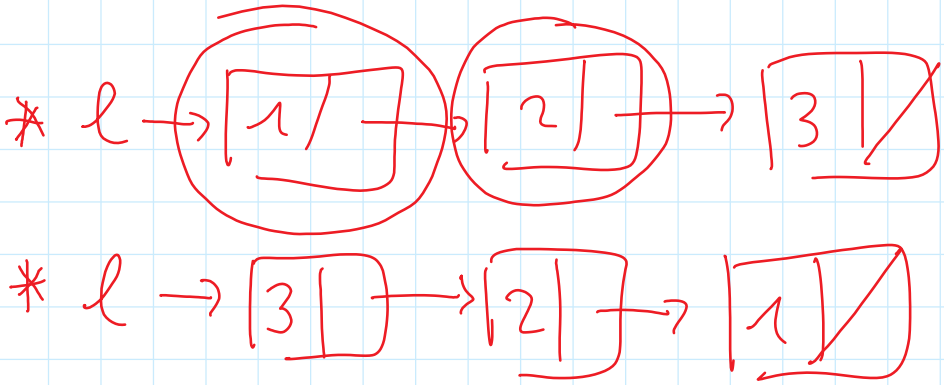




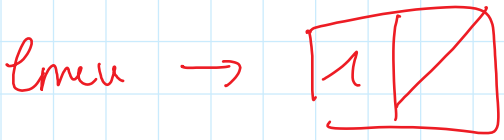
prec = cur -> next;

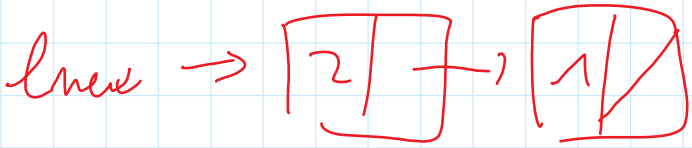
prec == hz

Rovesciare l'ordine degli elementi di una lista.



lnew == NULL

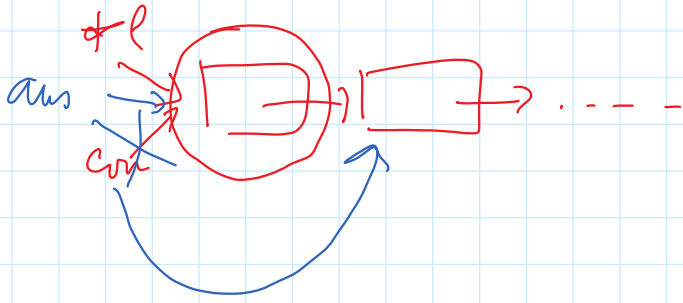




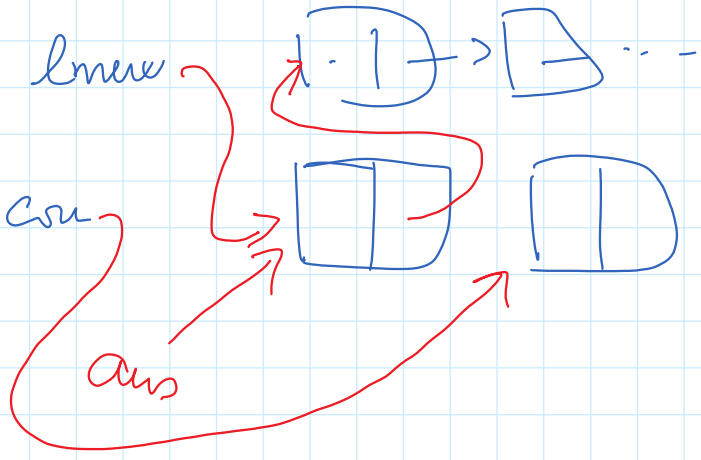
*l = lnew

```
void reverse (ElementoDiLista **l)
{ ElementoDiLista *lnew = NULL;
  ElementoDiLista *cur = *l;
  while (cur != NULL)
```

lnew == NULL




```
  { ElementoDiLista *aus = cur;
    cur = cur->next
    aus->next = lnew;
    lnew = aus;
  }
```




```
  *l = lnew;
}
```