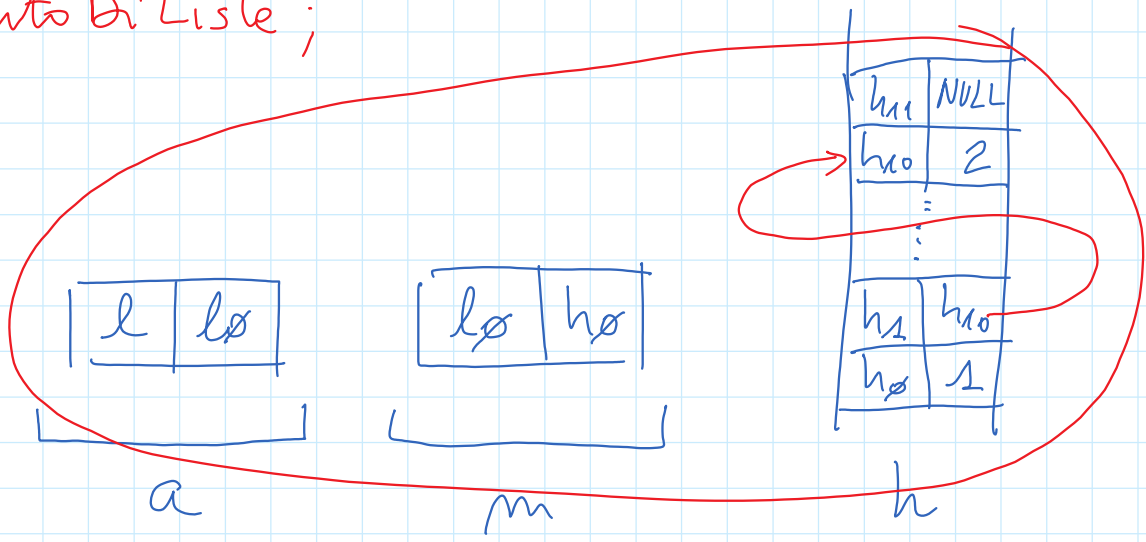
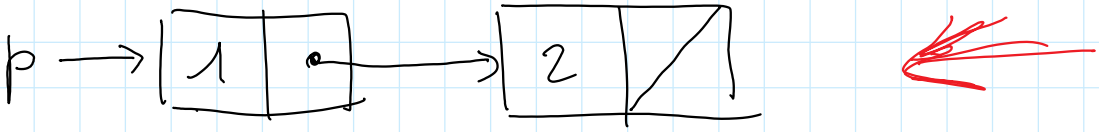
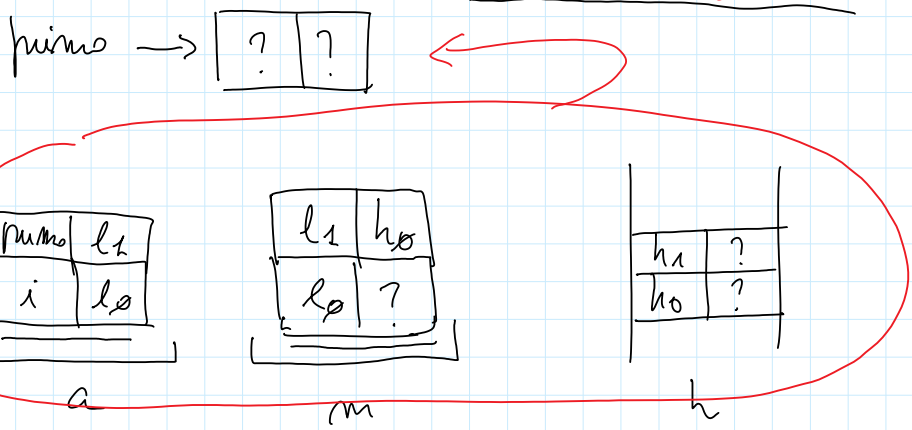
```
typedef
    struct el
        { int info;
          struct el * next;
        }
ElementoDiListe;
```



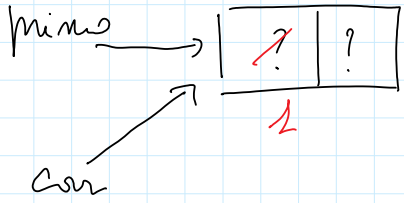


```
ElementoDiLista * crea (int n)        n ≥ 0
{ if (n == 0) return NULL;
  else
  { int i;
    ElementoDiLista * primo = malloc (sizeof (EDL));
```

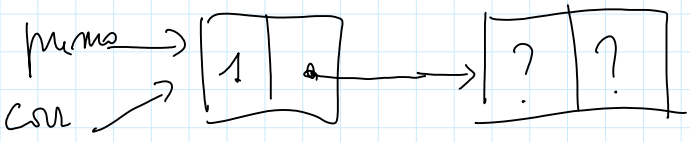
ElementoDiLista



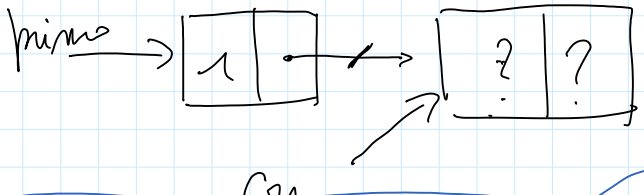

```
    ElementoDiLista * corr = primo;
```



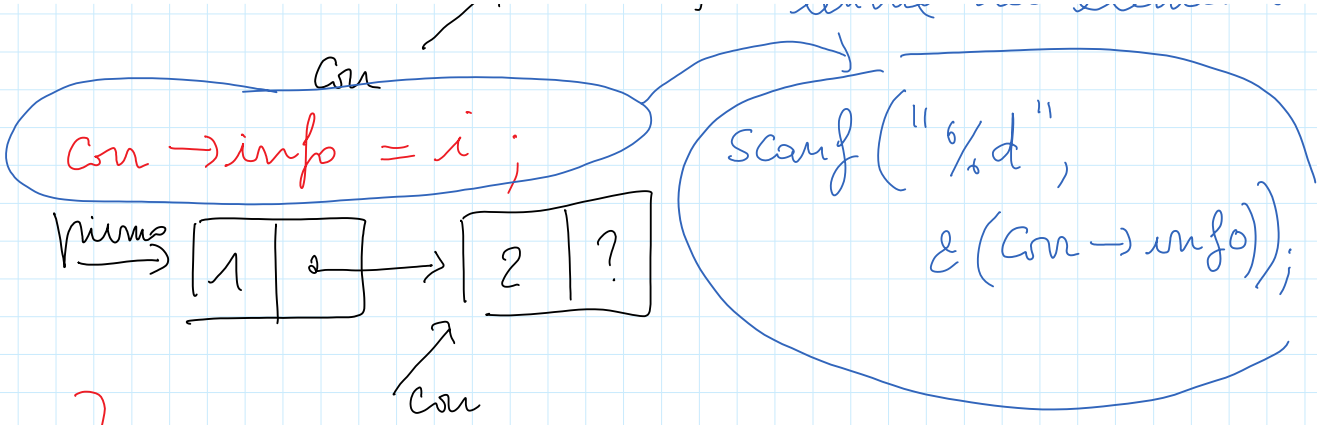

```
    corr -> info = 1;
    for (i=2; i<=n; i++)
    { corr->next = malloc (sizeof (EDL));
```

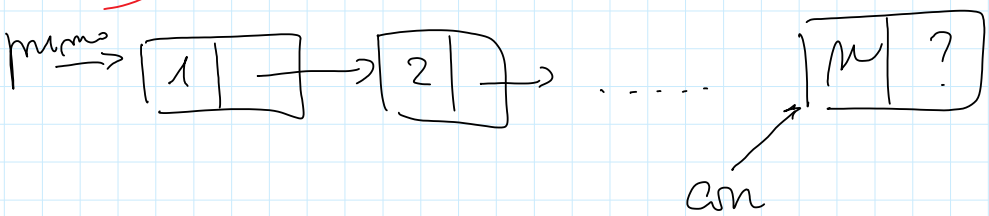



```
      corr = corr -> next;
```



lettura dell'elemento





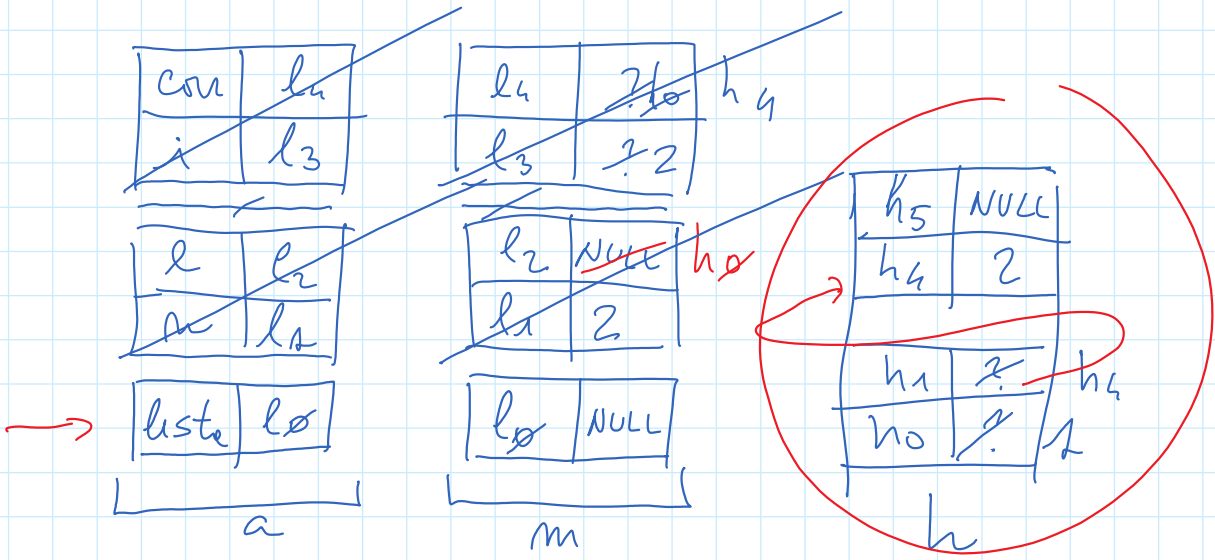

```
    con -> next = NULL;
    return primo;
  }
}
```

Vogliamo scrivere una procedura che crea una lista e modifica l'argomento con il puntatore al primo elemento della lista creata.

```
void crea (int n, ElementoDiLista * l)
{ if (n==0) l = NULL;
  else
  { int i;
    ElementoDiLista * corr;
    l = malloc (sizeof (EDL));
    corr = l;
    corr -> info = 1;
    for (i=2; i<=n; i++)
    { corr -> next = malloc (sizeof (EDL));
      corr = corr -> next;
      corr -> info = i;
    }
    corr -> next = NULL;
```

```
void crea (int n, ElementoDiListe * l)
{
   ...
}

main ()
{ ElementoDiListe * liste = NULL;
  crea (2, liste);
```



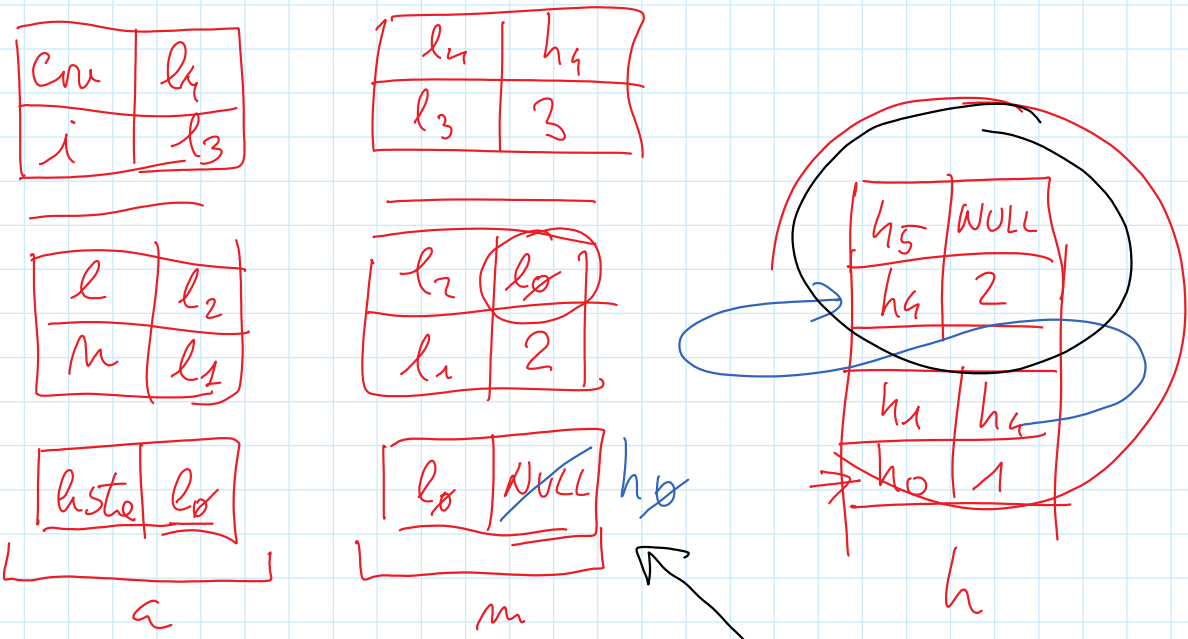
mercoledì 23 ottobre 2019 11:54

11:47

```
void crea (int n, ElementoDiLista ** l)
{
...
}
```

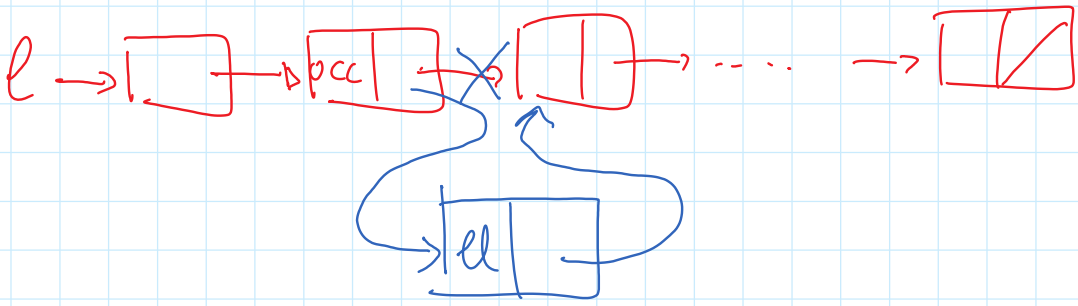
```
void crea (int n, ElementoDiLista ** l)
{ if (n==0) *l = NULL;
  else
  { int i;
    ElementoDiLista * cor;
    *l = malloc (sizeof (EDL));
    cor = *l;
    cor -> info = 1;
    for (i=2; i<=n; i++)
    { cor -> next = malloc (sizeof (EDL));
      cor = cor -> next;
      cor -> info = i;
    }
    cor -> next = NULL;
  }
```

```
}
}
main()
{ ElementoDiLista * lista = NULL;
  crea(2, &lista);
```

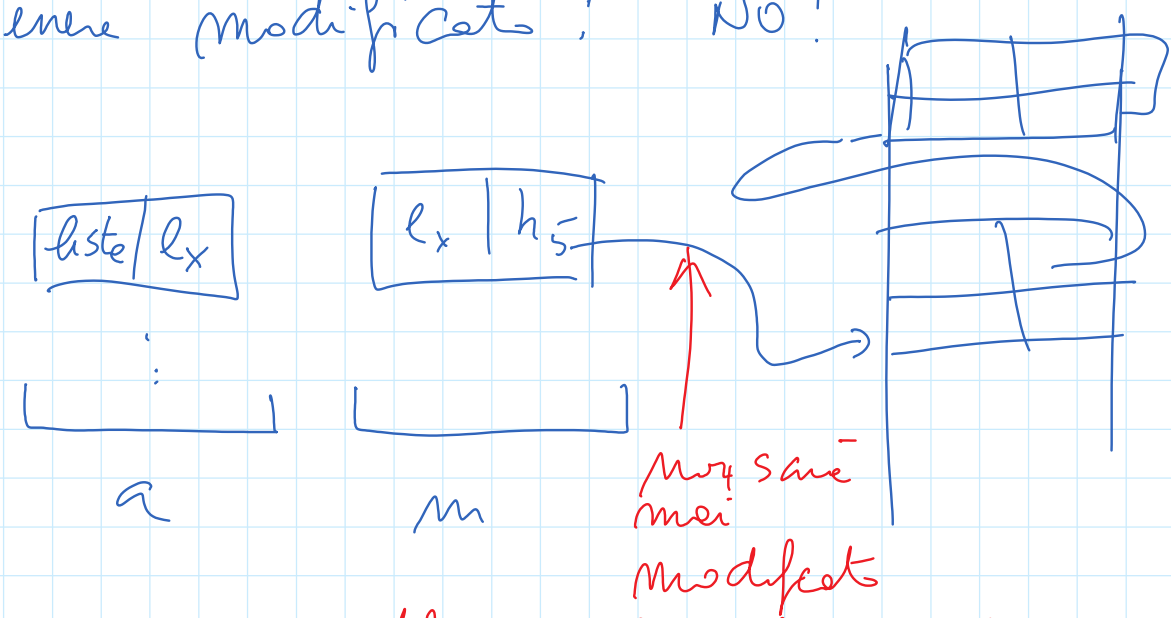


Il tipo ElementoDiLista ** va usato quando si deve modificare il puntatore al primo elemento della lista, come in questo caso

Scrivere una procedura C che inserisca in una lista un valore el dopo la prima occorrenza del valore occ. Se occ non compare nella lista la procedura la lascia invariata.

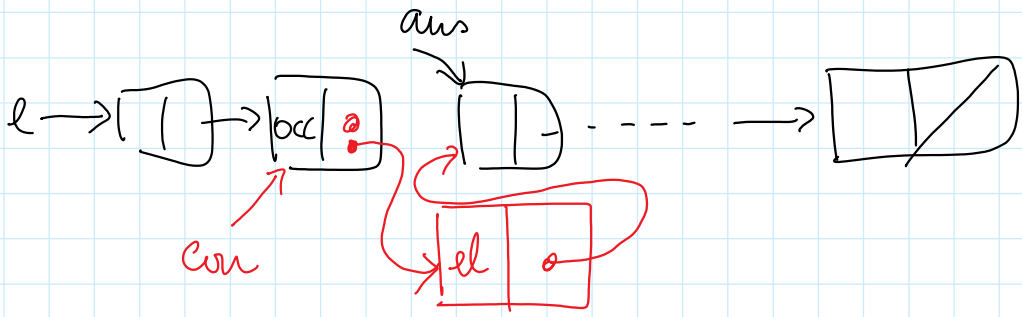


Il puntatore all'inizio della lista può essere modificato? NO!



(non sarà mai modificato il valore della variabile lista)

```
void insafter(int occ, int el, ElementoDiLista * l)
{
```




```
int trovato = 0;
ElementoDiLista * corr = l;
while (corr != NULL && !trovato)
    if (corr->info == occ) trovato = 1;
    else corr = corr->next;

if (trovato)
{ ElementoDiLista * aus = corr->next;
  corr->next = malloc(sizeof(EDL));
  corr->next->next = aus;
  corr->next->info = el;

}
}
```

```
corr = corr->next;
corr->next = aus;
corr->info = el;
```

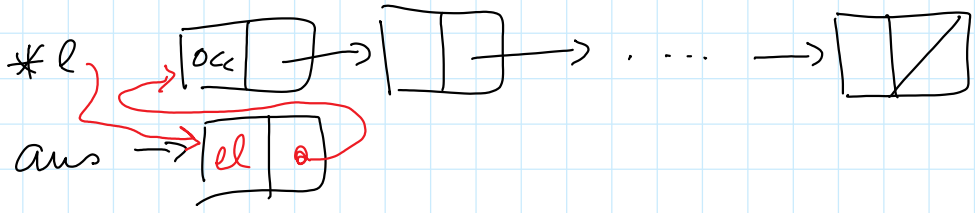
}

cur->info = el;

Scrivere una procedura C che inserisca un valore el, prima della prima occorrenza del valore occ.
Se la lista non contiene il valore occ rimane invariata.

Il puntatore al primo elemento della lista può essere modificato? SI!

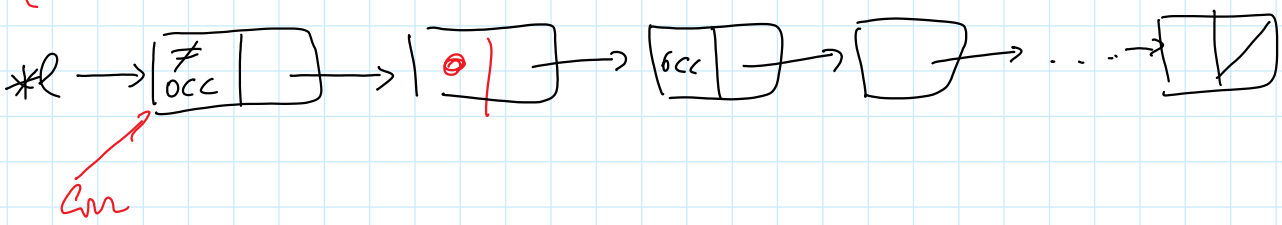
```
void insbefore(int occ, int el,
               ElementoDiLista **l)
{ if (*l != NULL)
    if (*l->info == occ)
    { ElementoDiLista *aus = malloc(sizeof(EDL));
```



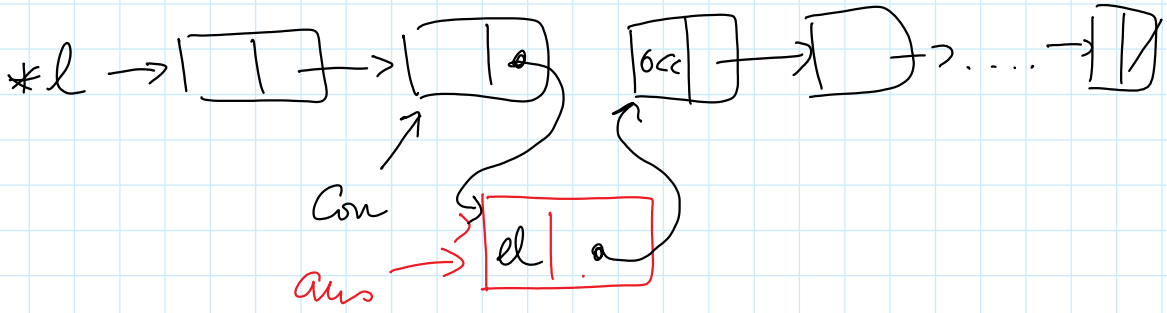

```
      aus->info = el;
      aus->next = *l;
      *l = aus;
    }
```

el

}



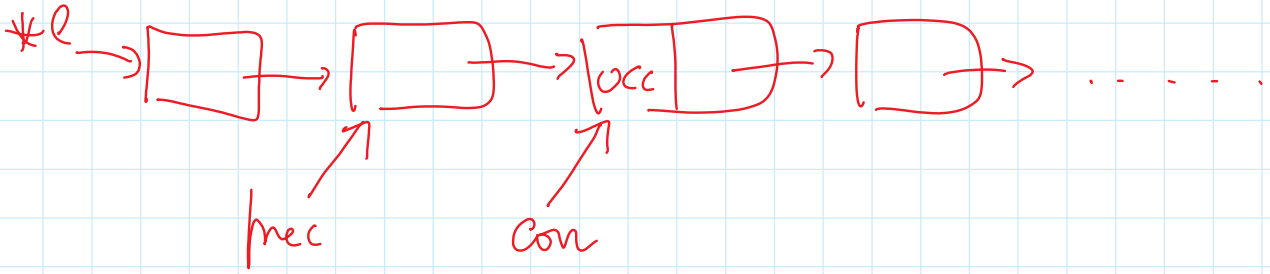

```
ElementoDiLista * curr = *l;
int trovato = 0;
while (curr->next != NULL && !trovato)
    if (curr->next->info == occ) trovato = 1;
    else curr = curr->next;
if (trovato)
{
```




```
ElementoDiLista * aus = malloc(sizeof(EDL));
aus->next = curr->next;
curr->next = aus;
aus->info = el;
```

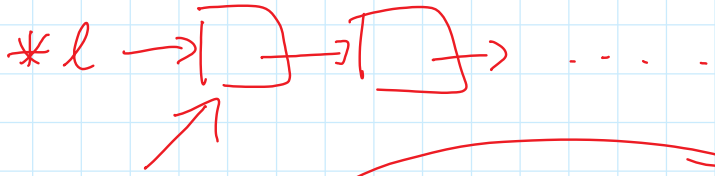
→ `curr->next->info = el;`

}



inizialmente

prec = NULL




```
prec = corr;
corr = corr -> next;
```

facciamo avanzare i due puntatori in modo che prec punti all'elemento che precede quello puntato da corr