

Scrivere una procedura che cancella da una lista il primo elemento = el. Se non ci sono elementi uguali a el la lista rimane invariata.

```
void conc (ElementoDiLista ** l, int el)
{
    ElementoDiLista * prec = NULL, con = *l;
    int trovato = 0;

```

```
    while (con != NULL && ! trovato)
        if (con->info == el) trovato = 1;

```

```
        else {
            prec = con;
            prec = prec->next;
            con = con->next;
        }

```

if (trovato)

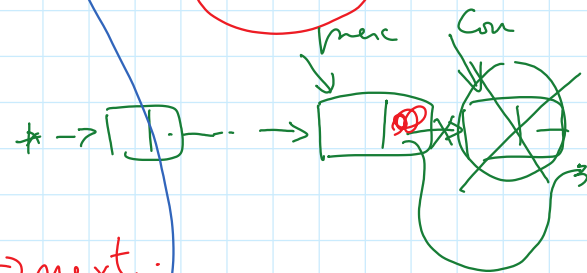
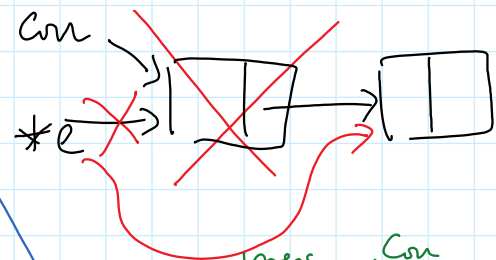
```
    if (prec == NULL)
    {
        *l = (*l)->next;
        free(con);
    }

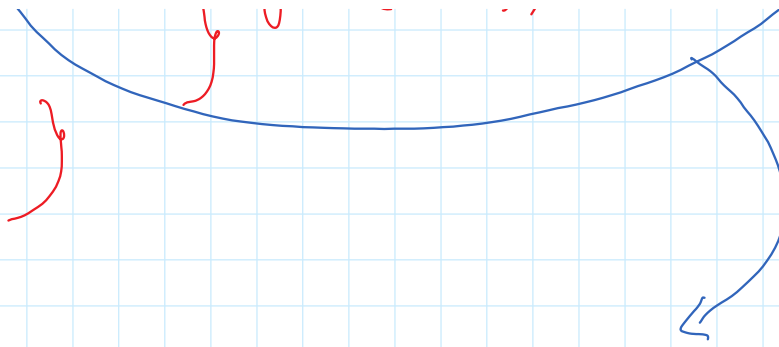
```

```
    else {
        prec->next = con->next;
        free(con);
    }

```

prec == NULL





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

Scrivere una procedura che cancella da una lista tutti gli elementi = a el
void conc (ElementiDi lista **l, int el)

```
{ ElementiDi lista * prec = NULL, con = *l;
```

```
while (con != NULL)
```

```
if (con -> info == el)
```

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

```
{ *l = *l -> next;
```

```
free (con);
```

```
con = *l;
```

```
}
```

```
else
```

```
{ prec -> next = con -> next;
```

```
free (con);
```

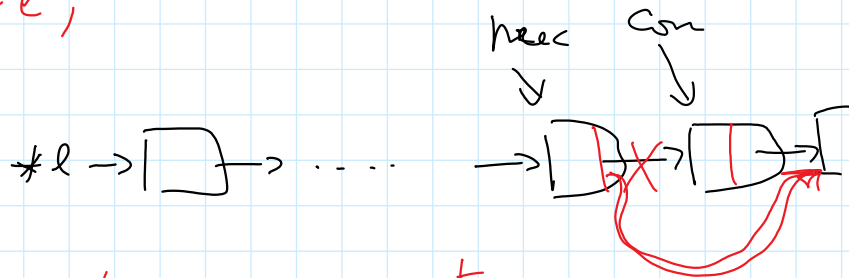
```
con = prec -> next;
```

```
}
```

```
prec = con;
```

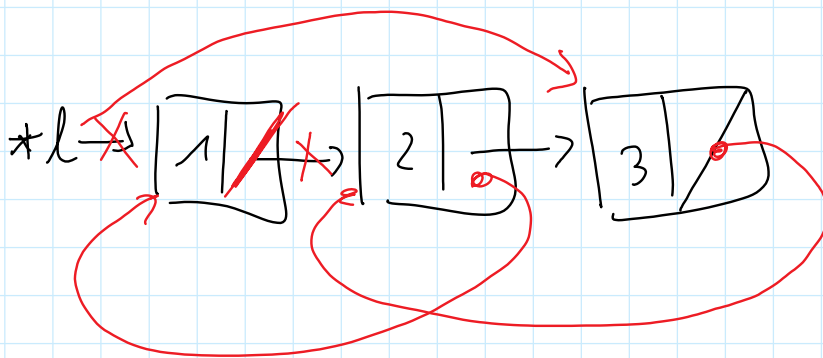
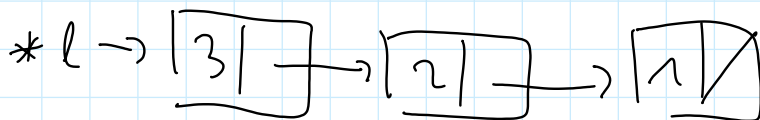
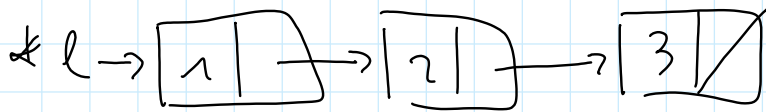
```
con = con -> next;
```

```
}
```



}

Scrivere una procedura che inverta l'ordine degli elementi di una lista



lnew == NULL

```
void reverse ( ElementiListe ** l )
{
    ElementiListe * lnew = NULL, lold = *l;
    while ( lold != NULL )
    {
        ElementiListe * ans = lold;
        lold = lold -> next;
        ans -> next = lnew;
        lnew = ans;
    }
}
```

*l = lnew;

}

