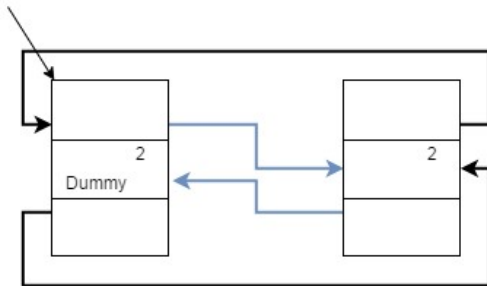


## Aufgabe 2.4

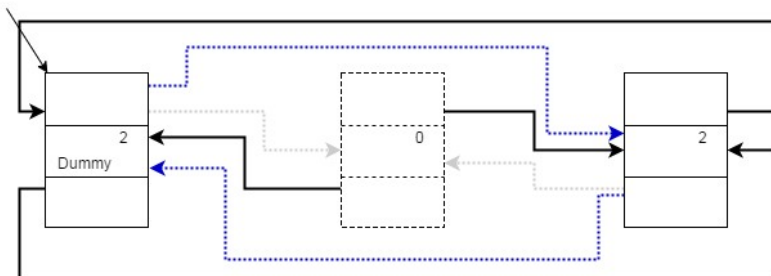
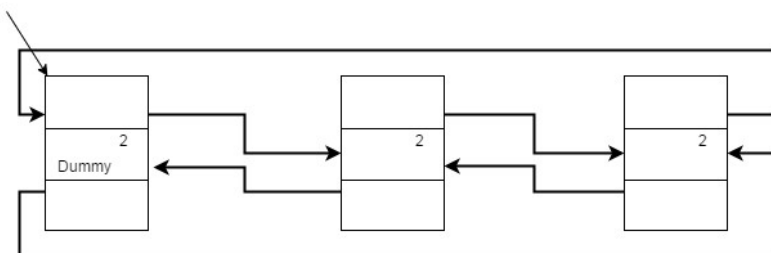
**Fall 1:** kein Knoten zwischen From und To.

Zwei Referenzen werden gelöscht und wieder erzeugt (blau), also keine Änderung.



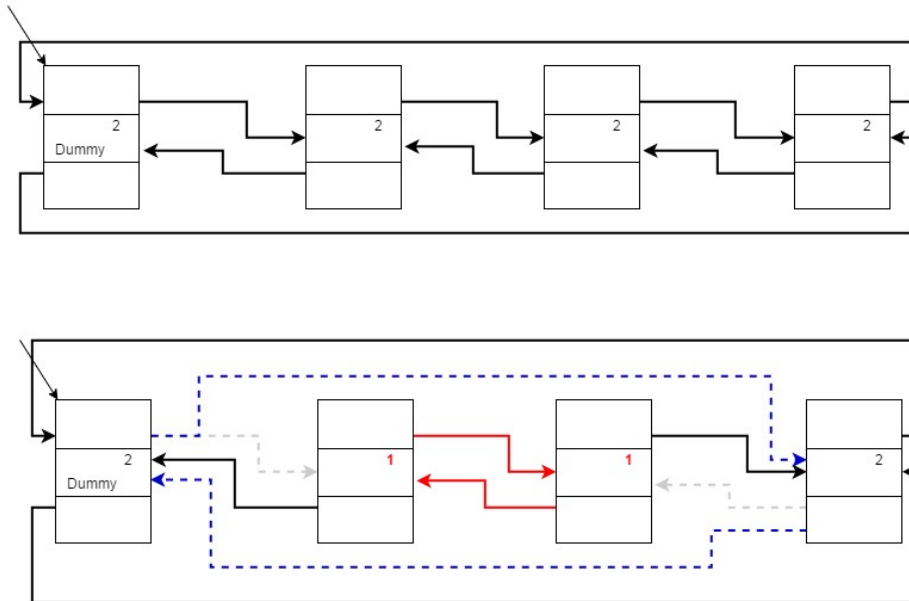
**Fall 2:** ein Knoten zwischen From (links) und To (rechts).

Am Anfang gibt es zwei Referenzen auf das mittlere Element. Beide Referenzen werden gelöscht (grau), und das Element darf auch gelöscht werden, da kein anderes Objekt Referenzen darauf hat.



**Fall 3:** mehrere Knoten zwischen From (links) und To (rechts).

Obwohl die Referenzen von From/To auf die "gelöschten" Elemente nicht mehr existieren, haben diese Elemente selbst Referenzen aufeinander, und deshalb dürfen sie nicht deallokiert werden. Solche „tote“ Ketten von Referenzen bleiben im Speicher, werden aber nicht mehr benutzt (eine *memory leak*).



Um das zu vermeiden, kann man die Referenzen zwischen den entfernten Knoten löschen:

```

procedure Remove_Between (From : in DLL; To : in DLL) is
    Temp, TempNext : DLL;
begin
    Temp := From;
    Loop_of_Destruction :
        while Temp.Forward /= To loop
            TempNext := Temp.Forward;
            Temp.Forward := Null;
            Temp := TempNext;
        end loop Loop_of_Destruction;
    From.Forward := To;
    To.Backward := From;
end Remove_Between;
    
```