



**T2Q1 (s, q) :**

This function modifies the circular linked list. Here's what it does:

1. It starts from the node *s*.
2. Traverses the list until it finds the node that comes BEFORE node *q*
3. Updates the `next` pointer of this node to point back to *s*.

Let's use the example where *Aptr* points to B and *Bptr* points to D:

For **T2Q1 (Aptr, Bptr) :**

1. Start at B (*s* = *Aptr* = B)
2. Traverse until we find the node before D (which is C)
3. Make C point back to B
4. Result: B → C → B becomes one circle

For **T2Q1 (Bptr, Aptr) :**

1. Start at D (*s* = *Bptr* = D)
  2. Traverse until we find the node before B (which is A)
  3. Make A point back to D
  4. Result: D → E → A → D becomes another circle
- One circular list starts at *Aptr* and includes nodes up to (and including) the node BEFORE *Bptr*. This means B → C → B forms the first circle.
  - Another circular list starts at *Bptr* and includes nodes up to (and including) the node BEFORE *Aptr*. This means D → E → A → D forms the second circle.