

Q.2] 1) Insertion at start

Algo insert start (SPART)

Begin {

1. IF $AVAIL = NULL$

PRINT ("No node available memory")

2. ELSE $AVAIL \rightarrow NEWNODE = AVAIL$ and $NEWNODE \rightarrow DATA = VALUE$

3. SET $PTR = SPART$

4. WHILE ($PTR \rightarrow NEXT \neq \overset{SPART}{NULL}$)

SET $PTR = PTR \rightarrow NEXT$

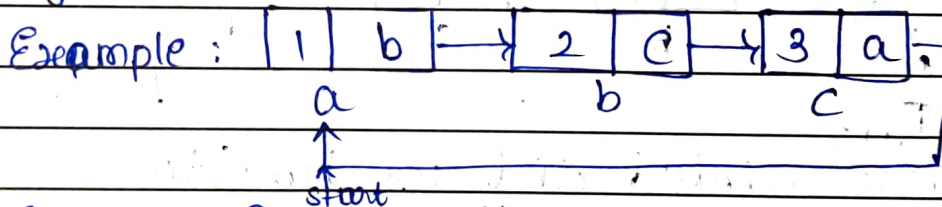
5. SET $PTR \rightarrow NEXT = NEWNODE$

6. SET $NEWNODE \rightarrow NEXT = SPART$

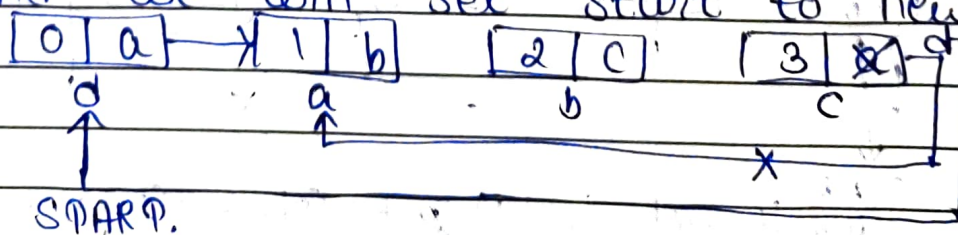
7. SET $SPART = NEWNODE$

8. RETURN $SPART$

} END.



Suppose I have the above circular linked list and I want to insert $[0 |]$. According to the algorithm pt 3, there will ptr variable that will traverse the list and as soon as it reaches the last node with next address as start, it will replace that with newnode address d and newnode next will have address of first node and we will set start to newnode :



In this way we have our node inserted in start.

2] Algo Delete last (START)

Begin?

1. ~~SET~~ ^{If} PTR = START = NULL

PRINT ("No node to delete")

2. ELSE SET PTR = START

3. WHILE (PTR → NEXT ≠ START

SET PREP = PTR

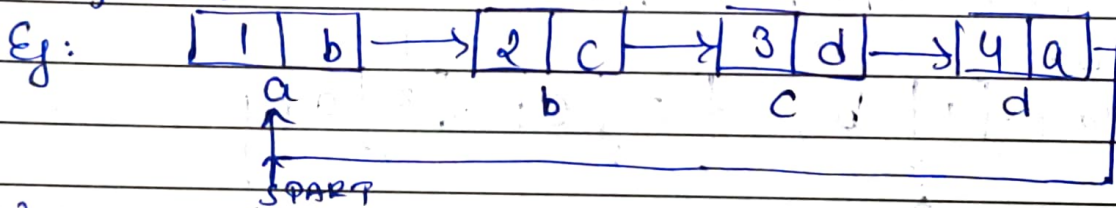
GET PTR = PTR → NEXT

4. SET PREP → NEXT = START

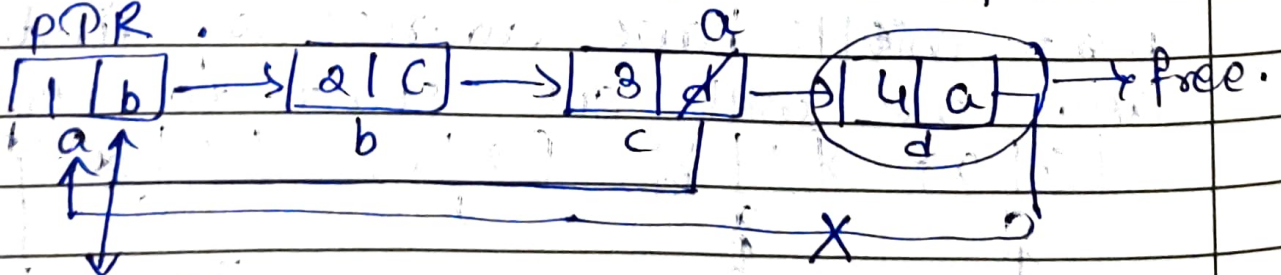
5. FREE (PTR)

6. RETURN START

} END.



Using above linked list we have to delete [4 | a]. We set two pointers ptr and prep. PTR goes ahead and PREP follows by keeping behind PTR. We traverse using PTR and set PREP according. Once we reach [4 | a] our PREP is on [3 | d]. We change its address to start and free the space of PTR.



∴ New linked list is [1 | b] → [2 | c] → [3 | a].