Doubly Circular Linked List – This is a special type of linked list in which every node contains 2 links. One to predecessor node and another to subsequent node. This helps to traverse the Linked List in either direction i.e. Left to Right OR Right to Left. Also note that the last node contains a link to the 1st node of LL and the 1st node contains a link to the last node.

1024 18 956

956 13 600

600 -25 1024

prev data next

head = 600 956 1024

struct node

{

int data;

struct node \*prev, \*next;

};

1st node Subsequent Nodes

600 18 600

956 18 956

600 18 600

newnode = 600 head = 600 newnode = 956

newnode->next = newnode; temp = 600

newnode->prev = newnode;

head = temp = newnode; newnode->prev = temp;

temp->next = newnode;

newnode->next = head;

head->prev = newnode;

How to traverse to last node of DBCL?

No need to write while loop.

temp = head;

while (temp->next != head)

temp = temp->next;

**OR**

temp = head->prev;