**IMPLEMENTATION OF QUEUE USING LINKED LIST**

**AIM:**

To write a python progam to implement queue using linked list.

**ALGORITHM:**

Queue operations:

**enQueue()** -This operation adds a new node after rear  and moves rear  to the next node.

**deQueue()** -This operation removes the front node and moves front to the next node.

**Display()-** Prints the queue list.

Appending node to the queue (enqueue):

1. Firstly, allocate the memory for the new node

2.Append the queue by adding an element to the end of the queue. The new element will be the last element of the queue.

3. If we insert element into an empty queue, the condition **front = None** becomes true. Now, the new element will be added as the only element of the queue and the next pointer of front and rear pointer both, will point to NONE.

4. If the queue contains more than one element, **front = None** becomes false. Now, we need to update the end pointer rear so that the next pointer of rear will point to the new node. Since, this is a linked queue, hence we also need to make the rear pointer point to the newly added node. We also need to make the next pointer of rear point to NULL.

Deletion of node from the queue (dequeue):

Deletion operation removes the element that is first inserted among all the queue elements.

1.Firstly, we need to check either the list is empty or not. The condition front = NULL becomes true if the list is empty, in this case, we simply print underflow.

2.Otherwise, we will delete the element that is pointed by the pointer front. For this purpose, copy the node pointed by the front pointer into the pointer ptr.

3.Now, shift the front pointer, point to its next node and free the node pointed by the node.

Display the nodes (Traversing): Displaying all the nodes of a stack needs traversing all the nodes of the linked list organized in the form of queue. For this purpose, we need to follow the following steps.

1. Copy the head pointer into a temporary pointer.
2. Move the temporary pointer through all the nodes of the list and print the value field attached to every node.