**Aim:**

Write a program to implement a Queue using Linked List.

**Algorithm:**

**Step 1 – Start**

**Step 2 –**Define functions enQueue, deQueue, display.

**Step 3 -**Define a '**Node**' structure with two members **data** and **next**.

**Step 4 -**Define two **Node** pointers '**front**' and '**rear**' and set both to **NULL**.

**Step 5 –**Display the menu of list of operations and make suitable function calls to perform user selected operation.

**Step 6 – Under enQueue, get value from user and** create a dynamically allocated **Node** pointer **newNode**.

Step 7 - set '**newNode → data**' to **value** and '**newNode → next**' to **NULL**.

**Step 8 -**Check whether queue is **Empty** (**front** == **NULL**)

**Step 9 -**If it is **Empty** then, set **front** = **newNode** and **rear** = **newNode**.

**Step 10 -**If it is **Not Empty** then, set **rear → next** = **newNode** and **rear** = **newNode**.

**Step 11 – Under deQueue function,** check whether **queue** is **Empty** (**front == NULL**).

**Step 12 -**If it is **Empty**, then display **"Status: Queue is Empty!"** and terminate from the function

**Step 13 -**If it is **Not Empty** then, define a **Node** pointer '**temp**' and set it to '**front**'.

**Step 14 -**Set '**front** = **front → next**' and delete '**temp**' (**free(temp)**).

**Step 15 – Under display function,** check whether queue is **Empty** (**front** == **NULL**).

**Step 16 -**If it is **Empty** then, display **'Status: Queue is Empty!'** and terminate the function.

**Step 17 -**If it is **Not Empty** then, define a **Node** pointer **'temp'** and initialize with **front**.

Step 18 – while '**temp**' reaches to '**rear**' (**temp → next** != **NULL**), Display '**temp → data**' and set **‘temp’** to temp -> next.

**Step 19 -**Display '**temp → data** ---> **NULL**'.

Step 20 - Stop

**Result:**

The program was successfully executed and achieved the aim of the program.

**Output:**

**** ****