Aim: Implementation of Priority Queue using linked list.

Objective:

1. Understanding working of priority queue and its applications.

Theory:

A priority queue is a data structure in which each element is assigned a priority. The priority of the element will be used to determine the order in which the elements will be processed. Lower the number, Higher the priority!

Priority

0 -Highest 1 -High

2 -Medium

The general rules of processing are:

1. An element with higher priority is processed before an element with a lower priority.
2. Two elements with the same priority are processed on a first-come-first-served (FCFS) basis.

A priority queue can be thought of as a modified queue in which when an element has to be removed from the queue, the one with the highest priority is retrieved first. The priority of the element can be set based on various factors. Priority queues are widely used in operating systems to execute the highest priority process first.

When a Priority Queue is implemented using a linked list, then every node of the list will have three parts:

(a) Information or data part

(b) Priority number of the element, and (c) the address of the next element.

If we are using a sorted linked list, then the element with the higher priority will come before the element with the lower priority. (for processing it before)

Algorithm:

1) Inserting an element in a priority queue

When a new element has to be inserted in a priority queue, we have to traverse the entire list until we find a node that has a priority lower than that of the new element. The new node is inserted before the node with the lower priority. However, if there exist an element that has the same priority as the new element, the new element is inserted after that element based on on **FCFS Rule**.

2) Deleting an element from priority queue

Deletion is a very simple process in this case. The first node of the list needs to be deleted so that data of that node gets processed first.

Source code for the implementation:

(Write only important functions)