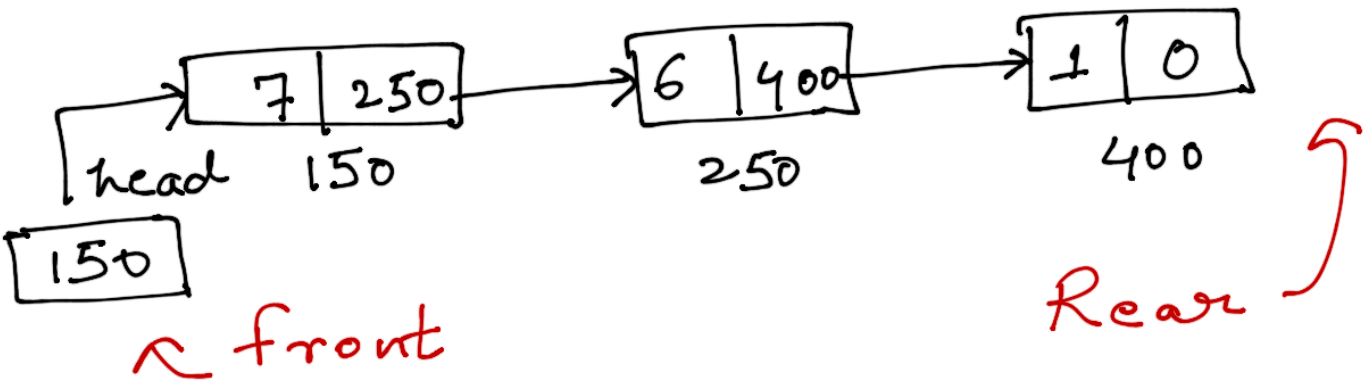
Implementation of Queues Using Linked Lists

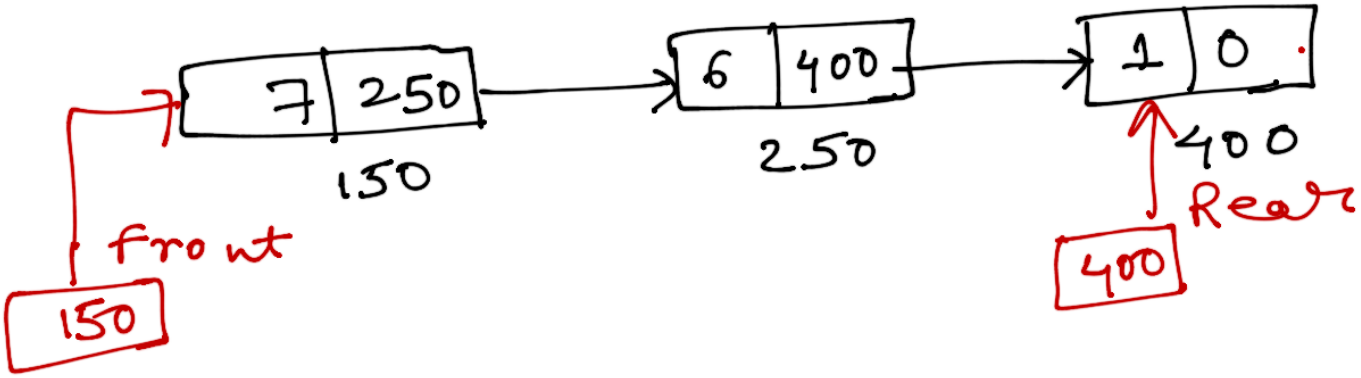


Queues

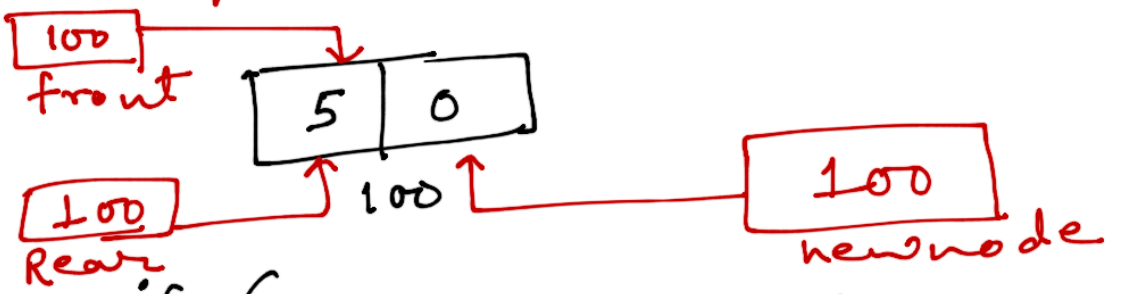
FIFO

Insertion - O(1)

Deletion - O(1)

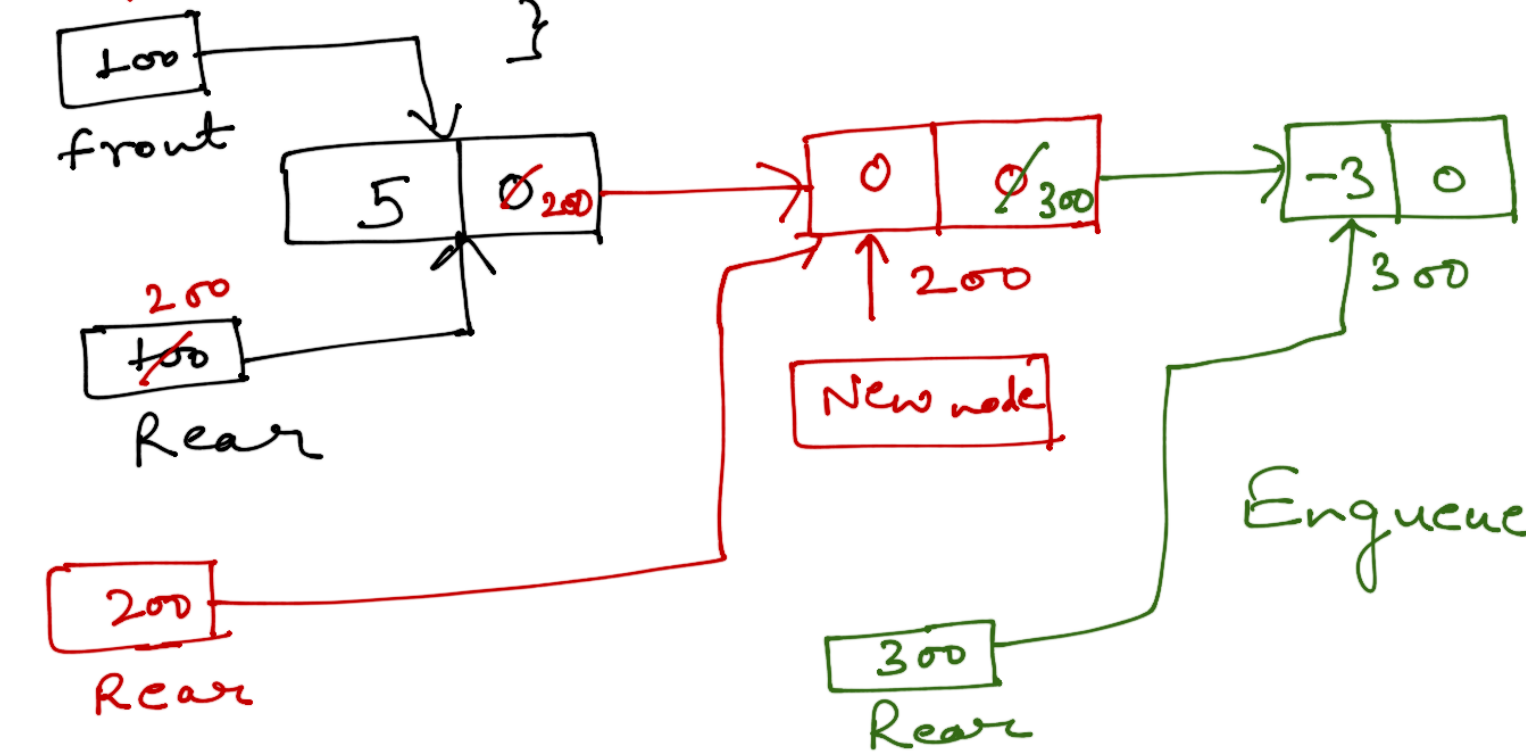


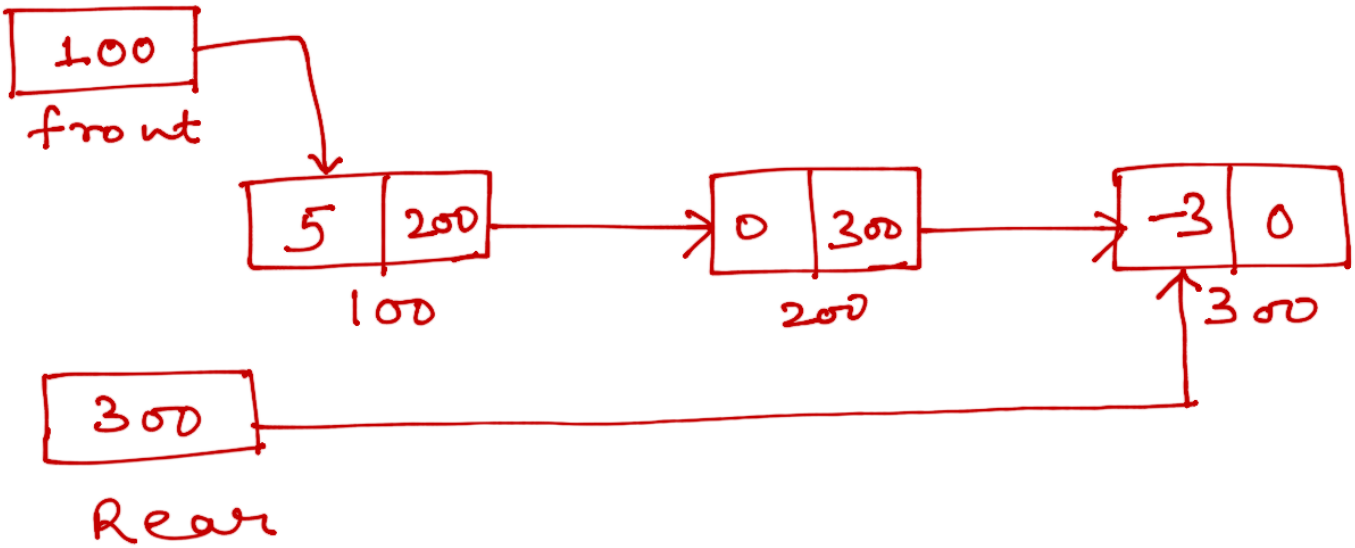
Enqueue(5)




```
if (front == 0 && rear == 0)
  { front = rear = newnode;
  }
else
  { rear -> next = newnode;
    rear = newnode;
  }
```

Enqueue(0)

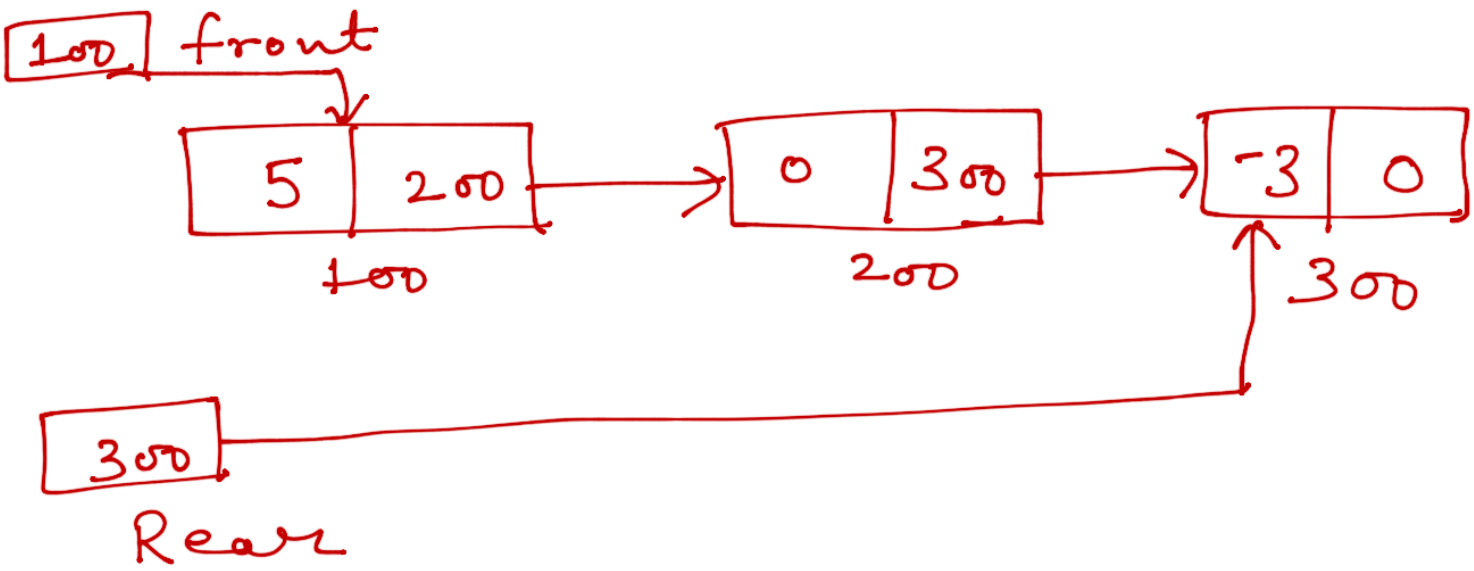




display()

```
if (front == 0 && rear == 0)
    {
Print "Queue is Empty"
    }
else
    {
        temp = front;
        while (temp != 0)
            {
                Print " temp -> data"
                temp = temp -> next;
    }       }
```

Dequeue



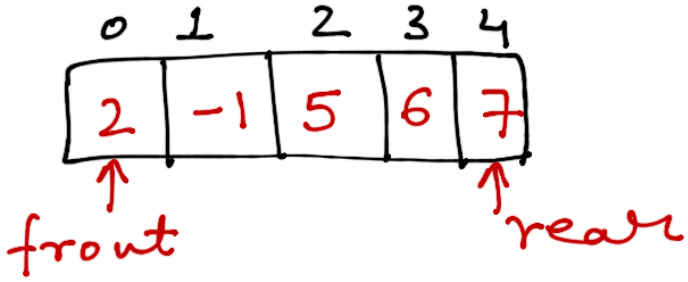
dequeue()

```
{   temp = front;
    if ( front == 0 & Rear == 0)
  {   PRINT "Queue is Empty"
  }
  Else      PRINT "Front → Data"
      { front = front → next
        free (temp)
  }
}
```

peek()

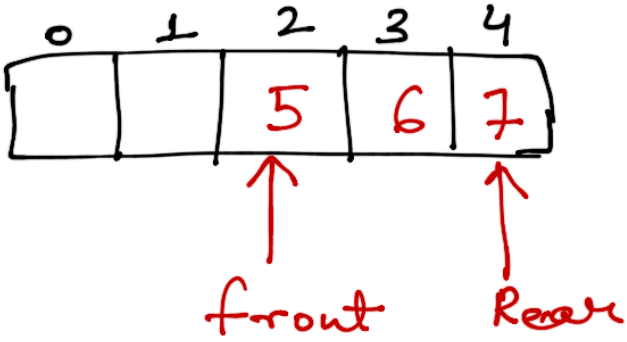
```
{
    if (front == 0 & Rear == 0)
        {PRINT "Queue is Empty"
        }
    else
        { PRINT " front -> Data"
        }
}
```

Circular Queue Using Arrays



front = 0

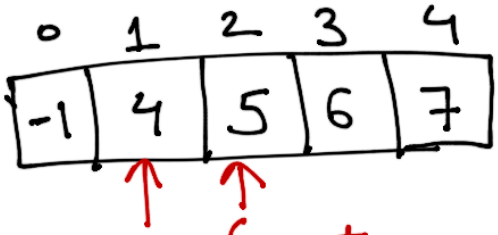
Rear = 4



Rear = size-1 ✗

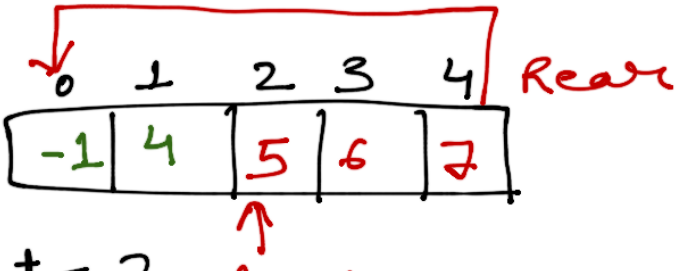
front = 2

rear = 4



Rear = 1

front = 2



front = 2

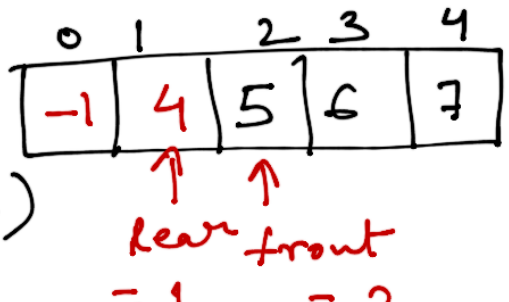
rear = 0

$$((rear + 1) \% N) == front$$

rear = 1

front = 2

N = 5 (size of Array)

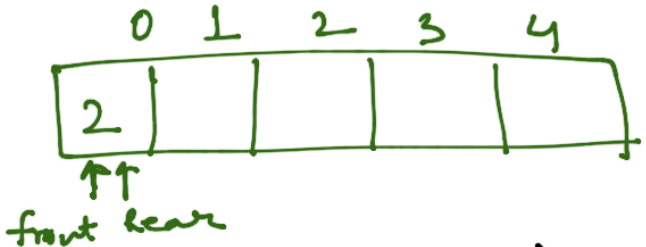


$$((1 + 1) \% 5) == 2$$

$$(2 \% 5) == 2$$

$$2 == 2$$ ✓

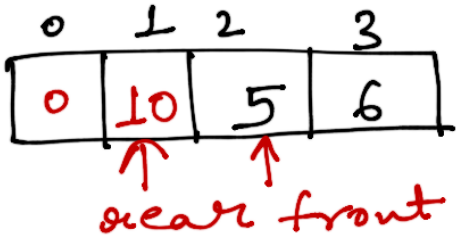
Enqueue (data)




```
{
    if (front == -1 && rear == -1)
    {   front = rear = 0
        queue [rear] = data
    }
    else if( ( (rear+1) % N) == front)
        {
          PRINT "Queue is Full"
        }
    else
        {   rear = (rear+1) % N
            queue [rear] = data
        }
}
```

Dequeue

```
{ if (front == -1 & rear == -1)
    {
      PRINT "Queue is Empty"
    }
  elseif ( front == rear)
    {  PRINT " queue[front] "
       front = rear = -1
    }
  else
    {
      front = (front +1) % N
    }
}
```

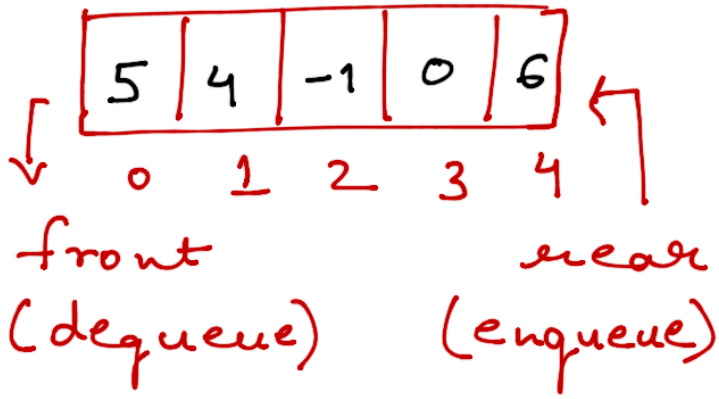


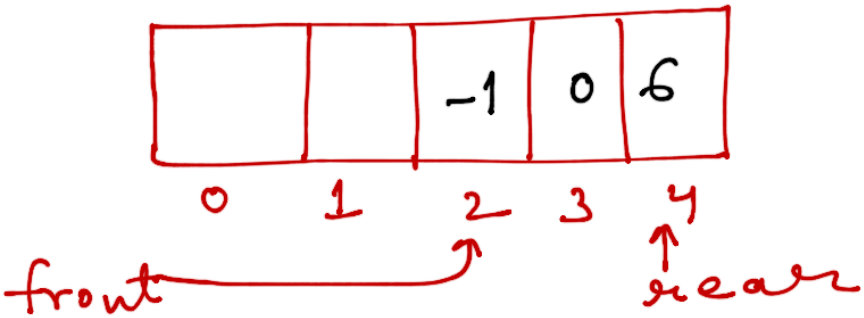

```
Display()
{
        i = front
  if (front == -1 & rear == -1)
     {
       PRINT "Queue is Empty"
     }
   else
     {
       while(i != rear)
         {
           PRINT queue[i]
             i = (i+1) % N
         }
          PRINT queue[i]
     }

}
```

Circular Queue Using Linked List

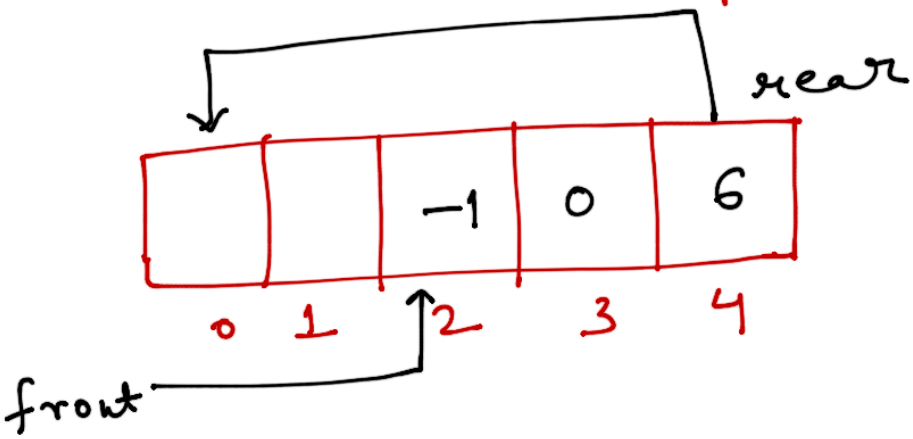
Dequeue()

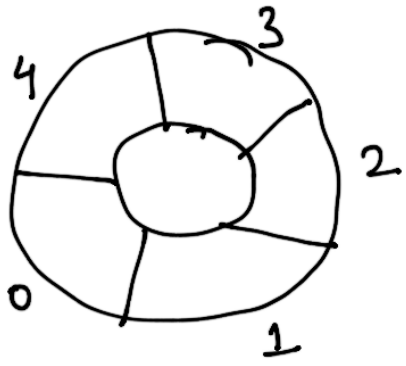
Dequeue()

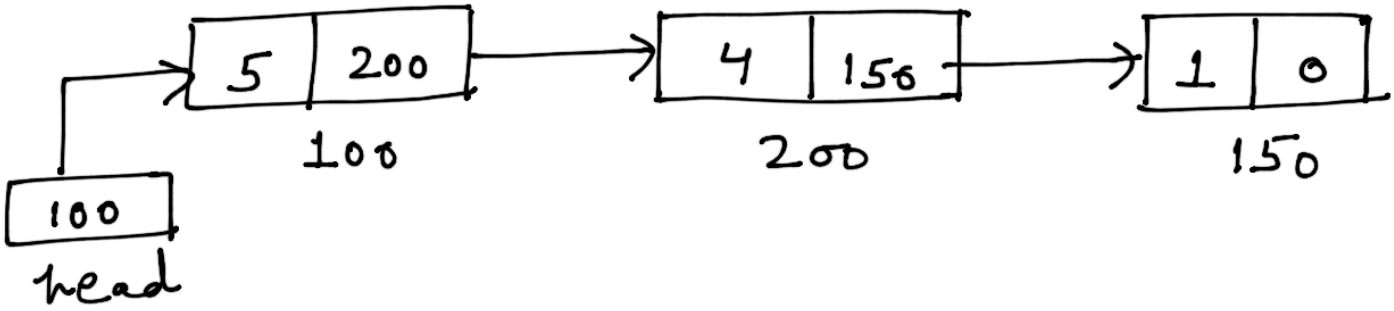




Can we Enqueue(5) here??

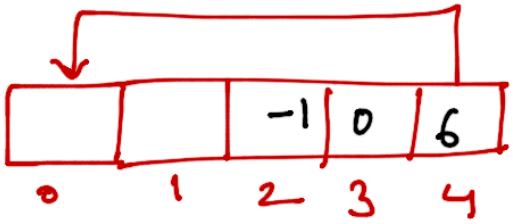




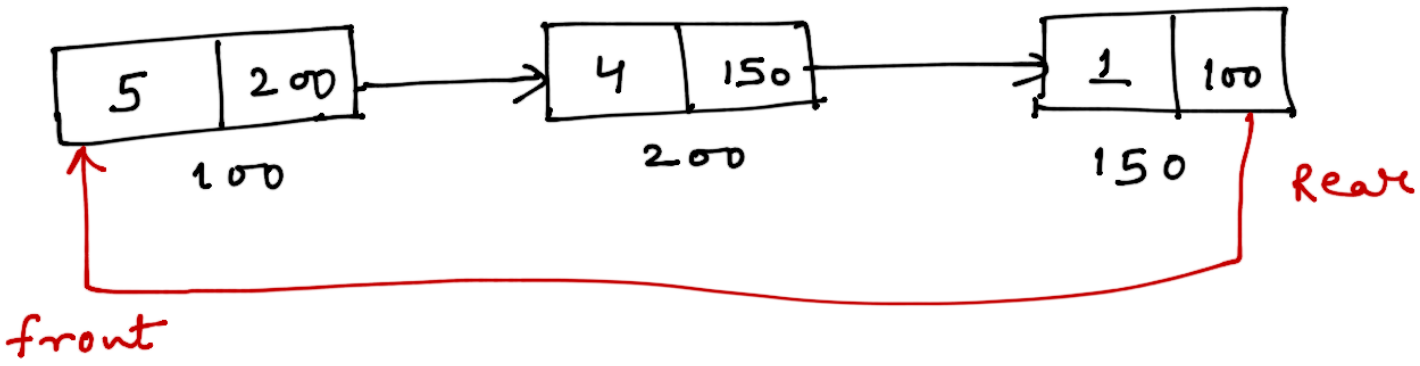


front
= 100

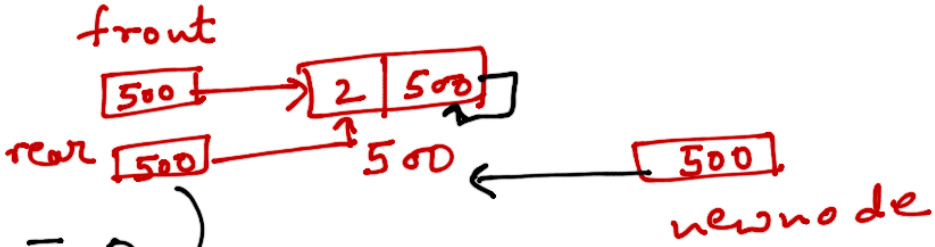
rear
= 150



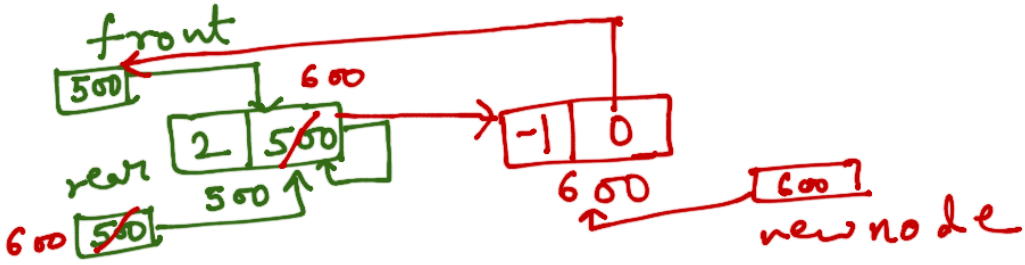
Circular Queue



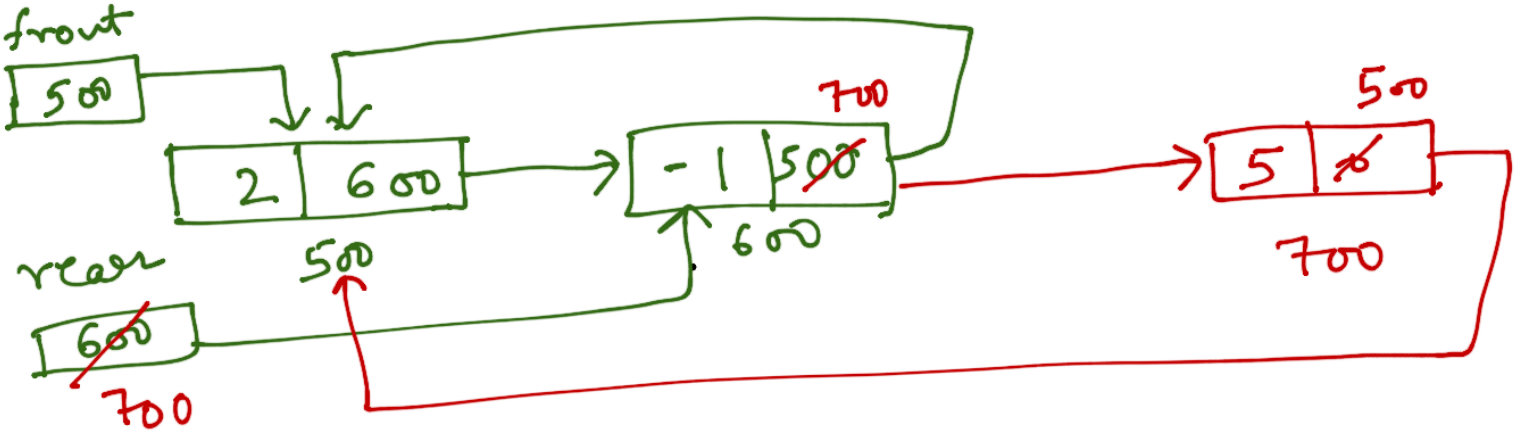
Enqueue



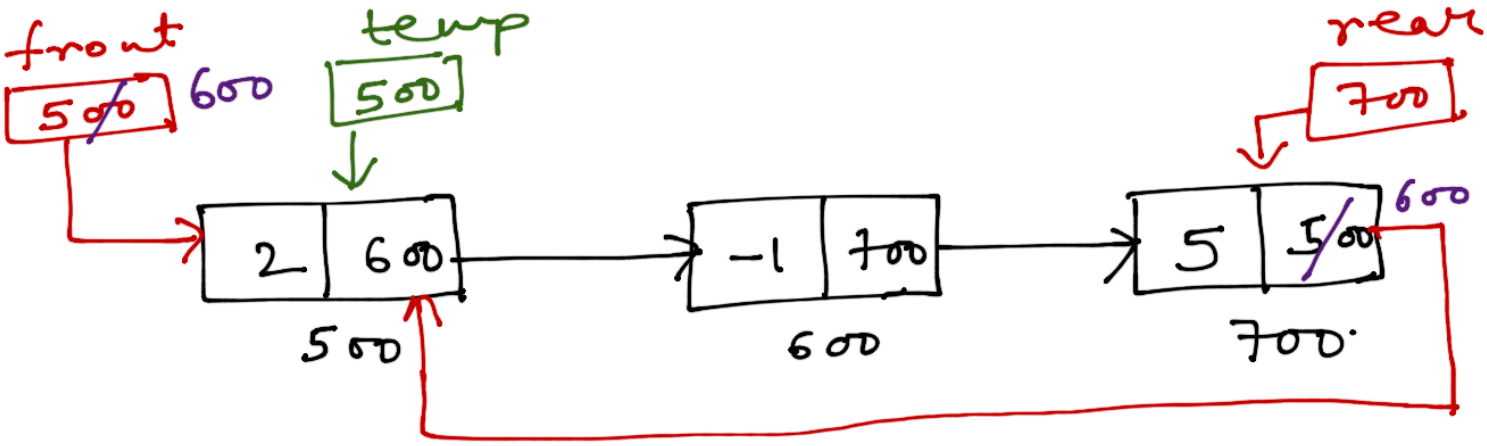

```
if (rear == 0)
{
    front = rear = newnode
    rear -> next = front
}
```




```
else
{
    rear -> next = newnode
    rear = newnode
    rear -> next = front
}
```

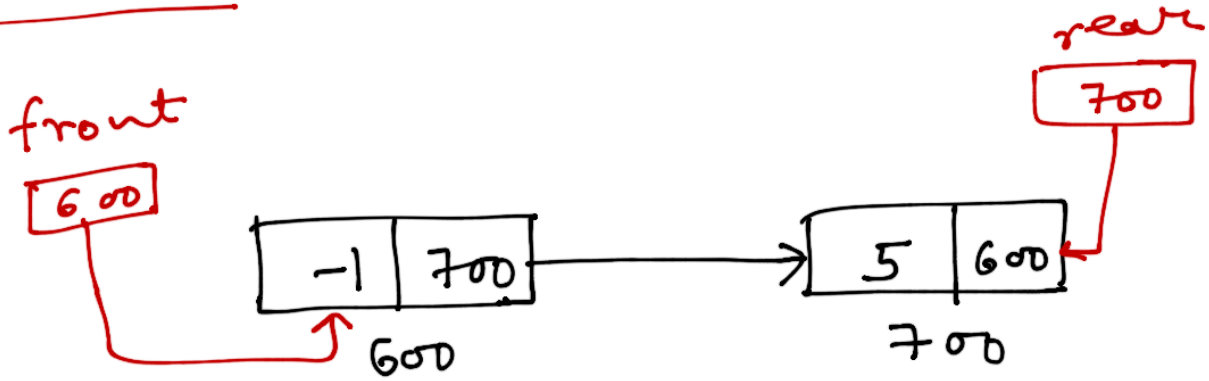


Dequeue ( )



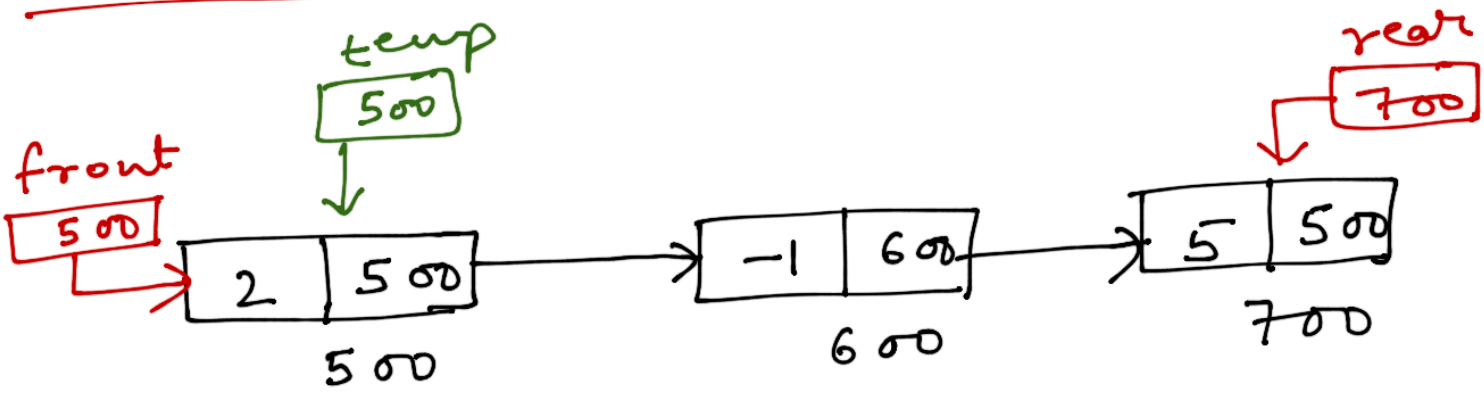

```
temp = front
if ( front == 0 & rear == 0 )
  { PRINT "Queue is Empty"
  }
elseif( front == rear)
    { front = rear = 0

                Free(temp)
    }
else
  {   front = front -> next
      rear -> next = front
  }   Free(temp)
```

peeK()




```
if (front == 0 & rear == 0)
    {
      PRINT "Queue is Empty"
    }
else
    {
      PRINT  front -> data
    }
```

display()




```
if (front == 0 & rear == 0)
  {
    Queue is Empty
  }
else
{     while( temp -> next != front)
      {
        PRINT  temp -> data
        temp = temp -> next
      }
}       PRINT temp -> data
```