```
1 Chranar Array Queue
class FullException(Exception):
      pass
class EmptyExceptionLException):
                                                    don't have are
class Array Queue:
            def -- (nit-- (suf, ele):
              suf-data = [data] * capacity
                                                      0-1000 440
              self capacity = capacity
              Suf. front = 0
              Suf. size = 0
           def enqueue(suf, ele):
                                                 Total = 3 bullium in
               if suf. size = = self. capacity:
                  raise Full Exception L'anene Overflow! ")
               rear = (self. front + self. size) 00 self. capacity
              self.data[rear]=ele
                                              hard flot - Holt burn
              Self. Sige + = 1
         det dequenelself):
                                                2(3haHrorens 23/26)
                               STOWN STANT TOOK TOWN IT
             if self. size == 0:
                 rouse Empty Exception ("Queue Underflow!")
             deleted Value = Self. data[self. front]
             self. data [self-front] = None
             self. front = (self. front + 1)° lo self. capacity
             Self-sige - =1
             return deleted Value
        def firstlself):
            if self. size == 0:
               raise Empty Exception ("Queue underflow 1,")
            ateturn self. data [self. front]
        def IsEmpty (suf):
             return self. size == 0
        def -- len -- Louts:
             return self. size
```

```
Class LinkedList Queue:

class Node:

def --init--iself, data=None, next=None):
```

```
timan transferrent in
     def -- init -- (self, data = None, next = None):
           suf-data = data
           suf-next = next
 def -- Init -- (self):
    self. head = None
    Suf. size = 0
 def enquerelself, etc!
     new Node= suf-Modelele, Mone)
         print (" could not create new Mode")
    if newNode == None;
     if suf-size ==0:
         suf-head = new Node
                                     11 111.50 - 501 coparty
         Suf-size +=1
                                (" ) was no sumper " ) i dry
          duturn
     current Node = suf-head
     tail Mode = suf-head
     while ( wrrenthode)!
             if wrent Node next == None;
                touil Node = current Node
                break
             else:
               ourrent Mode = current Mode. next
                                              11 30 F 3512 3 138 41
                              Color ( brancist violet page ) . The
     tailNode, next=numbode
                              Chord tis Joseph 1 12 marketis
     suf-size+=1
                    set ports (self front +1) . self coposity
def dequue (suf):
    If suf. head == None:
       print ("Queue underflow (")
   deleted Value = self-head. data
   self. head = self. head. next
                                  Just solver and north 1 1 1 16
    suf. size - = 1
    return deleted Value
def traverselucue (suf):
     current Node = self. head
                                    Small + 61 - Kear / Alab 310
     while (current Node):
        print ( urrent Node . data , end = " ")
        wrenthode = currenthode, nut
                                                 Chialphyodal 304
```

Class (Charges Degree

```
3 [Array] arcular Array Dequeue
class CerArray Deque:
          def -- (nit -- Csuf, capacity = 10):
             suf-data=[mone] * capacity
             Self-capacity = capacity
             suf-size = 0
             suf. front =0
         def addfirst (suf, ele):
                                                            wil. Frad Dore
            if selfisize == suf-capacity:
                   print ("Dequene Overflow (")
            suf. data [self. front - 1] = ele
                                                          arrived a salparacount di
            suf-front=(suf. front-1)°10 self. capacity
           self. size +=1
                                                              0 13615 1118 11
         def addLost(suf, ele):
            if suf-size == suf-capacity.
                 print ("Dequeue Overflow, ")
            rear = (self-front + self-size) % suf. capacity
           suf. data[rear] = ele
           self. size +=1
        def deletefirst (suf):
            if suf-size==0:
                print ("Dequeue underflow!")
                                                   Iboliona for a shoulding
            deleted value = suf-data[suf-front]
                                                               1-5 + 1212 33 15
            self. front= (self. front +1) 00 self. capacity
           suf.size - = 1
            return deleted Value
        def deletelast(self):
            if suf-size == 0!
               print (" Dequeue underflow [")
               return
             rear = (suf. front + suf. size) % self. capacity
             delited Value = self. data[rear-1]
            self. data Ercar-iJ= None
            suf-size= suf-size-1
             return deleted value
       def Is Empty (self):
             return self-size == 0
```

```
def first (suf):
     print ("Dequeue underflow!,")
  if suf-size ==0!
     ruturn
                                                   from Jan And
  return suf.data[suf.front]
def last (suf):
   if self. size == 0:
     print ("Dequeue underflow!")
     return
   rear = (self-front + self-size) olo self. capailty
   return suf.data[rear-1]
det -- len -- (suf):
                                                   ("rung") times
                                                          ALUETY.
     return self. size
                                           shoulders dans shoulder
                                            abolion - 1210 - Nos-112
                                                  Cintaria Line
                               per about factor about must
```

```
arcular Libert Queue
class Mode:
      def -- Init -- Cself, data = None, next = None):
           Self-data=data
           suf-next = next
class
       CQ
      det -- Init -- (self):
                                                  ( Contratt or Jessent It has
          suf-tail: None
          Suf. size = 0
                                       ptlenger $112 91 626 12-$122 + 1 1023 3132 1 2001
      def
           enqueue (self, uc):
        new Mode = Model ele)
        if new Mode = = Mone:
              print ("Error")
              nturn
       if suf-size == 0:
           new Mode. next = new Mode
          new Mode, next = self-tail. next
         suf.tail. next = nuoNode
       Self-tail= new Node
       solf-sizet=1
      def dequeue (suf):
            ff self-size = =0:
              print ("Empty")
              return
           head= suf-tail.next
            deletedvalue = head data
            if self-size ==1!
                self-tail= None
            Usi:
              Self-tail.next=head.next
            suf. size -= 1
            return deleted Value
       def Traverse (self):
            currentNode = self-tail-next
         for i in rangelsuf-size):
                  print ( turrent Node-data, end="")
                  surrent Node = current Node, next
```

(3112) - ast _ 3st

def first (self)! if suf-size == 0: paint ("Error") head= suf-tail.next return head data def is-Empty(suf): return self-size ==0 def -- len -- (self): return self-size det topesuf): nturn suf. tail. data def traverselself): aurent Node = self. tail. next for (in range (self-size): sazarozzassaham Wa) ahali a shalfam print (current Node, data, end = ") jours assortion, is ourrent Mode = current No de next shokesn: virg 192237 mg I were not been asked the all the state of the makes

```
6 Doubly Linked List
class Mode:
       def -- (net -- (self, data=None, prev=None, next=None):
             SUf. data=data
             suf. pru = prev
             Suf. next = next
                                                        O k or la file and or
class DLL:
      def -- init-- (self):
            Self. header = Node ()
            self. trailer = Node ()
            suf. header.next= self. trailer
            suf. trailer. prev = suf. header
            suf. sige = 0
            Insert_between (self, ele, predecessor, successor):
      def
             newNode = Node (ele, predecessor, sucressor)
             if newNode == None!
                                   The state their was thing
                 print (" would not create new Node")
                 return
             predecessor, next = newNode
             successor. prev = nuoNode
             self-size += 1
             return newNode
      def Forward Traversal (self):
           current Node = self. header. next
           for I in range (self-size):
                   print ( urrent Node. data)
                   current Node = current Node next
       det Backward Troversal (self):
           currentNode=self.trailer.prev
           for 1 in range (suf-size):
                   print (current Node, data)
                   eurentNode = ourrentNode, prev
        def InsertAtHead (suf, ue):
           return suf. Insert-between (ele, self. header, suf. header. next)
       def InsertAtTail (self, ele):
           return suf. Insert_ between(cle, suf. trailer, prev, suf. trailer)
```

DeleteGiven Nodel suf, node): def deleted Value = node. data predecessor = node . prev successor = node, next predecessor. next = successor successor. next = predecessor o estimations are prosented of open the predeusor = None successor = None suf-size -= 1 return deleted Value

super selbested to

(Jan man +) 155 30 boltwerp 3 13 15 () 102 110 113 15

() world being super Thomas

C'I wasterbrue ramed "It i'm

A Joseph man supre I Jales

manh-ten rebend- 1112 creater

Court Deant dider from

runn sul, mailer, prevenent

William SHE STAN

(when religion first should be the really

Casalenge 110 Should har

```
@ LinkedList Deque
class
      LinkedList Deque (DLL):
       def
            add first (self, ele):
          self. Insert-between cele, self-header, suf-header. next)
       def addlast(self, ele):
          self. Insert-between (ele, suf-trailer, prev, self-trailer)
       def deleteFirst(self):
          if self. size ==0:
              print ("Deque underflow !")
              return
          return seif. Delete GivenNode (self. header, next)
       def deletelast(self):
          it self-size == 0:
             print (" Deque underflow !, ")
             return
         return suf. Delete Giren Node (self-trailer prev)
       def first (self):
           if suf-size = = 0:
               print (" Deque under flow ! ")
              return
           return self. header. next. data
        def (ast (self):
           if suf. size = = 0'.
              print (" Deque underflax")
              return
          return suf, trailer, prev, next
        def is Empty (suf):
             ruturn suf-size == 0
        def -- len -- (self):
             return self. size
```