```python
from exceptions import Empty

class CircularLinkedList:
    class _Node:
        __slots__ = '_element', '_next'

        def __init__(self,element, next):
            self._element = element
            self._next = next

    def __init__(self):
        self._head = None
        self._tail = None
        self._size = 0
    def __len__(self):
        return self._size
    def is_empty(self):
        return self._size == 0

    def add_first(self,e):
        newest = self._Node(e,None)
        if self.is_empty():
            newest._next = newest
            self._head = newest
            self._tail = newest
        else:
            self._tail._next = newest
            newest._next = self._head
        self._head = newest
        self._size += 1

    def add_last(self,e):
        newest = self._Node(e,None)
        if self.is_empty():
            self._head = newest
            newest._next = newest
        else:
            newest._next = self._tail._next
            self._tail._next = newest
        self._tail = newest
        self._size += 1

    def add_any(self,e,pos):
        newest = self._Node(e,None)
        thead = self._head
        i = 1
        while i < pos:
```

```
48              thead = thead._next
49              i += 1
50          newest._next = thead._next
51          thead._next = newest
52          self._size += 1
53
54      def remove_first(self):
55          if self.is_empty():
56              raise Empty('Linked List Empty')
57          oldhead = self._tail._next
58          self._tail._next = oldhead._next
59          self._head = oldhead._next
60          self._size -= 1
61          if self.is_empty():
62              self._tail = None
63          return oldhead._element
64
65      def remove_last(self):
66          if self.is_empty():
67              raise Empty('Linked List Empty')
68          thead = self._head
69          i = 0
70          while i < len(self) - 2:
71              thead = thead._next
72              i += 1
73          self._tail = thead
74          thead = thead._next
75          self._tail._next = self._head
76          value = thead._element
77          self._size -= 1
78          return value
79
80      def remove_any(self, pos):
81          if self.is_empty():
82              raise Empty('Linked List Empty')
83          thead = self._head
84          i = 1
85          while i < pos-1:
86              thead = thead._next
87              i += 1
88          value = thead._next._element
89          thead._next = thead._next._next
90          self._size -= 1
91          return value
92
93      def display(self):
94          thead = self._head
```

```
 95          i = 0
 96          while i < len(self):
 97              print(thead._element, end='-->')
 98              thead = thead._next
 99              i += 1
100          print()
101
102 CL = CircularLinkedList()
103 CL.add_last(10)
104 CL.add_last(20)
105 CL.add_last(30)
106 CL.add_last(40)
107 CL.display()
108 print('Deleted: ', CL.remove_first())
109 CL.display()
110 CL.add_first(70)
111 CL.display()
112 print('Deleted: ', CL.remove_last())
113 CL.display()
114 CL.add_last(80)
115 CL.display()
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
```