Q1:-

While traversing the linked list we usually take a variable temp(temp which points to the list so that while traversing we don't loss the connection with the list) and updates it value to next ptr.

The common syntax goes like:-

temp=temp.next

The terminating condition of this loop is when the temp becomes null or None(depends on that language we are using). In case of circular linked list ,we know the tail ptr or the end node has the reference of the head node or the first node and so we it would be an infinite loop which we don't need. So we specify it by saying:-

If temp.next==temp.head:

    break

So if we need to know if we have reached the first element we will use the condition temp.next==temp.head.(or temp=temp.head)

To make us more clear temp here used is the instance of the class as defined below:

```
class Node:

    def __init__(self,value=None):# default constructer to define attributes of the object

            self.head=None

            self.tail=None

            self.value=value
```

Q2. What are the practical applications of a circular linked list? (Try to find applications in your respective fields)

- Circular Linked Lists can be used to manage the computing resources of the computer.
- Data structures such as stacks and queues are implemented with the help of the circular linked lists.
- Circular Linked List is also used in the implementation of advanced data structures such as a Fibonacci Heap.
- It is also used in computer networking for token scheduling.

- Round Robin scheduling technique in games.
- Audio/Video Streaming
- Circular Escalators