In [80]:

```
#Represents the node of list.
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None

class CreateList:
     #Declaring head and tail pointer as null.
    def __init__(self):
        self.head = Node(None)
        self.tail = Node(None)
        self.head.next = self.tail
        self.tail.next = self.head
        #self.dataadd = dataadd
    def add(self,data):
        newNode = Node(data)
        #Checks if the list is empty.
        if self.head.data is None:

        #If list is empty, both head and tail would point to new node.
            self.head = newNode
            self.tail = newNode
            newNode.next = self.head
        else:
            #tail will point to new node.
            self.tail.next = newNode
            #New node will become new tail.
            self.tail = newNode
            #Since, it is circular linked list tail will point to head.
            self.tail.next = self.head


    def search(self,element):
        current = self.head;
        i = 1;
        flag = False;
        #Checks whether list is empty
        if(self.head == None):
            print("List is empty");
        else:
            while(True):
                #Compares element to be found with each node present in the
                if(current.data ==  element):
                    flag = True;
                    break;
                current = current.next;
                i = i + 1;
                if(current == self.head):
                    break;
            if(flag):
                print("Element is present in the list at the position :  " +
            else:
                print("Element is not present in the list");


  #Displays all the nodes in the list
```

```
    def display(self):
        current = self.head
        if self.head is None:
            print("List is empty")
            return
        else:
            print("\nNodes of the circular linked list: ")
            #Prints each node by incrementing pointer.
            print(current.data)
            while(current.next != self.head):
                current = current.next
                print(current.data)

    #This function will add to the end of the list.
    def addAtEnd(self,data):
        newNode = Node(data);
        #Checks if the list is empty.
        if self.head.data is None:
            #If list is empty, both head and tail would point to new node.
            self.head = newNode;
            self.tail = newNode;
            newNode.next = self.head;
        else:
            #tail will point to new node.
            self.tail.next = newNode;
            #New node will become new tail.
            self.tail = newNode;
            #Since, it is circular linked list tail will point to head.
            self.tail.next = self.head;
    def delete(self, value):
        current = self.head
        prev = current.next
        while prev.next != self.head:
            prev = prev.next
        if current.data == value:
            prev.next = current.next

        current = current.next
        prev = prev.next
        while current.data != value and current != self.head:
            current = current.next
            prev = prev.next

        if current.data == value:
            prev.next = current.next

```

In [81]:
```
cl = CreateList();
i = True
print("Enter The Data Below To Perform CircularLinkedList or Enter Exit to End")
while i == i:
    a=input("Enter : ")
    if a.lower() == "exit":
        i=False
        break
    else:
        cl.add(a)

cl.display();
```

```
Enter The Data Below To Perform CircularLinkedList or Enter Exit to End
Enter : Awais
Enter : Ajmal
Enter : Arham
Enter : Aqib
Enter : Manohar
Enter : Apple
Enter : 3
Enter : 2.0
Enter : exit

Nodes of the circular linked list:
Awais
Ajmal
Arham
Aqib
Manohar
Apple
3
2.0
```

In [82]:
```
nsearch = input("Enter The Data To Find : ")
cl.search(nsearch)
```

```
Enter The Data To Find : Apple
Element is present in the list at the position :  6
```

In [83]:
```
print("Enter The Number to add in A List")
class CircularLinkedList:
    addinalist = input("Enter : ")
    #Adding 1 to the list
    cl.addAtEnd(addinalist);
    print("\nAdded Successfully")
    cl.display();
```

```
Enter The Number to add in A List
Enter : Mazz

Added Successfully

Nodes of the circular linked list:
Awais
Ajmal
Arham
Aqib
Manohar
Apple
3
2.0
Mazz
```

In [84]:
```
ndelete = input("Enter The Data To Delete From List : ")
cl.delete(ndelete)
```

```
Enter The Data To Delete From List : 3
```

In [85]:
```
print("\nAfter Deletion Process Data:\n")
cl.display();
```

```
After Deletion Process Data:


Nodes of the circular linked list:
Awais
Ajmal
Arham
Aqib
Manohar
Apple
2.0
Mazz
```

In [ ]:
```
#Done
```