Question :

There is a group of students playing in the playground and they made a circle by joining each other's hands. Help P.E.T Master separate them into two equal circular groups.

Constraints:

1 <= N <= 1000

Input Description:

Number of Students, N
N student's names

Output Description:

Two group of Student names each group in new line

Solution:

```
class Node:
  def __init__(self, data):
     self.data = data
     self.next = None

class CircularLinkedList:
  def __init__(self):
     self.head = None
  def push(self, data):
     ptr1 = Node(data)
     temp = self.head
     ptr1.next = self.head
     if self.head is not None:
        while (temp.next != self.head):
           temp = temp.next
        temp.next = ptr1
     else:
        ptr1.next = ptr1  # For the first node
     self.head = ptr1

  def printList(self):
     temp = self.head
     if self.head is not None:
        while (True):
           print("%d" % (temp.data))
           temp = temp.next
           if (temp == self.head):
```

```
                    break


    def splitList(self, head1, head2):
        slow_ptr = self.head
        fast_ptr = self.head
        if self.head is None:
            return
        while (fast_ptr.next != self.head and
               fast_ptr.next.next != self.head):
            fast_ptr = fast_ptr.next.next
            slow_ptr = slow_ptr.next
        if fast_ptr.next.next == self.head:
            fast_ptr = fast_ptr.next
        head1.head = self.head
        if self.head.next != self.head:
            head2.head = slow_ptr.next
        fast_ptr.next = slow_ptr.next
        slow_ptr.next = self.head

head = CircularLinkedList()
head1 = CircularLinkedList()
head2 = CircularLinkedList()

head.push(12)
head.push(56)
head.push(2)
head.push(11)

print("Original Circular Linked List")
head.printList()

# Split the list
head.splitList(head1, head2)

print("\nFirst Circular Linked List")
head1.printList()

print("\nSecond Circular Linked List")
head2.printList()
```

Test Cases:

Test Case 1:

Input:
```
6
2 5 9 1 4 7
```
Output:
```
2 5 9
1 4 7
```

Test Case 2:
Input:
```
2
2 5
```
Output:
```
2
5
```

Test Case 3:
Input:
```
7
A B C G H R E
```
Output:
```
A B C
G H R E
```

Test Case 4:
Input:
```
3
2 9 7
```
Output:
```
2 9
7
```

Test Case 5:
Input:
```
10
2 5 9 1 4 7 0 3 6 8
```
Output:
```
2 5 9 1 4
7 0 3 6 8
```