Swap pairs of elements in list

You are given a singly linked list:

```
struct node
{
    int n;              // value of element
    struct node *next; // pointer to next element in list
}
```

Write a function that will swap pairs of elements in a given singly-linked list. Note that you have to actually swap the elements, not just the values, and that you should modify the list in place (i.e. you should not create a copy of the list).

```
struct node *swapPairs(struct node *l);
```

For instance, the list 1→2→3→4→5→6… becomes 2→1→4→3→6→5…

IMPORTANT: Your implementation must also work for circular lists where the tail is pointing back to the head of the list. You do not have to check if the tail points to an intermediate (non-head) element.

If you would like to solve the problem in C#, consider the following definitions:

```
public class Node
{
     public int n;     // value of element
     public Node next; // pointer to next element in list
}

public Node swapPairs(Node head);
```