Pairwise Swap Nodes of linked list

Given a singly linked list, write a function to swap elements pairwise.

> *Input : 1->2->3->4->5->6->NULL*
> *Output : 2->1->4->3->6->5->NULL*
>
> *Input : 1->2->3->4->5->NULL*
> *Output : 2->1->4->3->5->NULL*
>
> *Input : 1->NULL*
> *Output : 1->NULL*

For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5.

Method 1 : Swapping Data

C++   **Java**

```java
import java.io.*;
import java.lang.*;
import java.util.*;

class Node {
    int data;
    Node next;
    Node(int x)
    {
        data = x;
        next = null;
    }
}

class Test {
```

```java
        Node head = new Node(1);
```

```java
        head.next.next.next.next = new Node(5);
        printlist(head);
        pairwiseSwap(head);
        printlist(head);
    }

    static void pairwiseSwap(Node head)
    {
        Node curr = head;
        while (curr != null && curr.next != null) {
            int temp = curr.data;
            curr.data = curr.next.data;
            curr.next.data = temp;
            curr = curr.next.next;
        }
    }

    public static void printlist(Node head)
    {
        Node curr = head;
        while (curr != null) {
            System.out.print(curr.data + " ");
            curr = curr.next;
        }
        System.out.println();
    }
}
```

Output:

```
1 2 3 4 5
2 1 4 3 5
```

Menu

Time Complexity - O(n)

Method 2: **Changing Pointers/References**

▦
All

```java
 import java.io.*;
import java.lang.*;
import java.util.*;

class Node {
    int data;
    Node next;
    Node(int x)
    {
        data = x;
        next = null;
    }
}

class Test {

    public static void main(String args[])
    {
        Node head = new Node(1);
        head.next = new Node(2);
        head.next.next = new Node(3);
        head.next.next.next = new Node(4);
        head.next.next.next.next = new Node(5);
        printlist(head);
        head = pairwiseSwap(head);
        printlist(head);
    }

    static Node pairwiseSwap(Node head)
    {
        if (head == null || head.next == null)
            return head;
        Node curr = head.next.next;
        Node prev = head;
        head = head.next;
        head.next = prev;
```

📖
Articles

▶
Videos

</>
Problems

❓
Quiz

🎗
Contest

Menu

Track Progress

§ 90% Money-Back! Iplete. (72%)

Courses

Tutorials

Jobs

Practice

Contests

All

Articles

Videos

Problems

Quiz

Contest

```java
        prev.next = curr;
        return head;
    }

    public static void printlist(Node head)
    {
        Node curr = head;
        while (curr != null) {
            System.out.print(curr.data + " ");
            curr = curr.next;
        }
        System.out.println();
    }
}
```

Output:

```
1 2 3 4 5
2 1 4 3 5
```

Time Complexity – O(n)

Mark as Read

🐞 Report An Issue

If you are facing any issue on this page. Please let us know.

Menu

Track Progress

95 of **132** Complete. (72%)