https://course.acciojob.com/idle?question=b78231a6-7180-4990-9601-dd839afbdf6f

- EASY
- Max Score: 30 Points
-
-

Reverse Alternate Nodes of a Singly Linked List

For a given Singly Linked List of integers, you are required to reverse alternate nodes and append them to the end of the list.

Input Format

The first line of input contains an integer `N` denoting the length of linked list.

The following line contains `N` space separated integers denoting the elements of the linked list.

Output Format

Print a single line containing space-separated integers denoting the elements of the resultant linked list.

Example 1

Input

```
5
2 4 6 8 10
```

Output

```
2 6 10 8 4
```

Explanation

The given linked list is 2->4->6->8->10 then after reversing the alternate nodes we will have 2->6->10->8->4.

Assuming 0 based indexing, odd indexed nodes are the alternate nodes that is, in the given linked list the node with value 4 and the node with value 8 are the alternate nodes.

List without alternate nodes: 2->6->10

List with alternate nodes: 4->8

Reversing the list with alternate nodes: 8->4

After appending the reversed alternate nodes at the end, the updated list will be 2->6->10->8->4.

Example 2

Input

```
3
-10 20 -1
```

Output

```
-10 20 -1
```

Explanation

Since ,there is no alternate nodes, output list is same as input.

Constraints

`1 <= N <= 10^5`

`1 <= DATA <= 10^6`

Topic Tags

- **Linked lists**

My code

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

class Node
{
    int data;
    Node next;

    Node(int data, Node next)
    {
        this.data = data;
        this.next = next;
    }

    Node() {}
}

public class Main
{

  static void display(Node h)
  {
    Node p=h;
    while(p!=null)
     {
        System.out.print(p.data+" ");
        p=p.next;
     }
  }


  static Node reverse(Node a)
     {
        Node dummy = new Node();
        Node tail = dummy;
        while (true)
        {
          if (a == null) break;
        Node d=a;
         a=a.next;
         d.next=tail.next;
         tail.next=d;

        }

        return dummy.next;
    }




static Node rev(Node a)
  {
```

```
 if(a!=null)
 {
  Node r=null,q=null ,t=null,p=null,h=null;
    r=a;
   p=a.next;
   q=p;

 while(true)
  {
  if(r.next ==null)
          break;
          else
  {
   r.next=q.next;
     r=r.next;
  }

    if(q.next==null)
            break;
       else   {
     q.next=r.next;
     q=q.next;
          }
   }
   h=reverse(p);
     //    display(h);
         Node no=a;
while(no.next!=null)
        no=no.next;
         no.next=h;//add secand part
```

```
      return a;
    }
    return a;
  }

        public static void main (String[] args) throws
java.lang.Exception
        {
                //your code here
      Scanner s=new Scanner(System.in);
            int n=s.nextInt();
      Node a=null;
      int arr[]=new int[n];
for(int i=0;i<n;i++)
  arr[i]=s.nextInt();
 for(int i=n;i>0;i--)
   a=new Node(arr[i-1], a);
 a=rev(a);
  display(a);


        }
}
```