https://course.acciojob.com/idle?question=c6f18527-8007-4ca5-b928-b9fc2db5f28f

- MEDIUM
- Max Score: 40 Points

# Middle Node Of Linked List

**Given the head of a linked list, return the middle node of the linked list.**

**Linked List Structure:**

```
public class Node
{
    int data;
    Node next;
    Node(int d) {data = d; next = null; }
}
public class LinkedList
{
    Node head;
    Node tail;
}
```

Note:- You just need to complete midpointOfLinkedList() function and return the mid point. The driver code will print the second half of the linked list.

Example:

If your list is `[5, 4, 3, 2]`, **the function should return the node at index 2, i.e. value 3.**

## Input Format

**You will be provided with an integer `n`, the number of elements in the linked list.**

**The next `n` integers denote the values of the nodes in the linked list.**

## Output Format

**Print the linked list starting from the middle node.**

## Example 1

Input

```
4
5 4 3 2
```

Output:

```
3 2
```

## Example 2

Input

```
3
5 7 1
```

Output:

```
7 1
```

## Constraints

```
1 <= n <= 10^5
```

**Topic Tags**

- **Linked lists**

# My code

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

```
import java.io.*;

public class Main
{
	public static void main (String[] args) throws
java.lang.Exception
	{
		//your code here
		Scanner sc=new Scanner(System.in);
		int n=sc.nextInt();
		linkedList l1=new linkedList();
		for(int i=0;i<n;i++)
			{
				l1.add(sc.nextInt());
			}
		   l1.delete(n);
		l1.print();
	}
}
class Node
	{
		int val;
		Node next;
		Node(int val)
		{
			this.val=val;
			this.next=null;
		}
	}
class linkedList
```

```
{
	Node head;
	Node tail;
	public void add(int val)
	{
		Node n=new Node(val);
		if(head==null)
		{
			head=n;
			tail=n;
			return;
		}
		tail.next=n;
		tail=n;
	}
	 public void delete(int n)
	{
		int count=1;
		int mid=(n/2)+1;
		Node temp=head;
		while(count!=mid)
			{
				temp=temp.next;
				count++;
			}
		tail=temp;
	}
	public void print()
	{
		Node temp=tail;
```

```
            while(temp!=null)
                {
                    System.out.print(temp.val+" ");
                    temp=temp.next;
                }
        }
}
```